APPLE II 繪圖遊戲系列①

施威銘 校訂
林和賢 編著

APPLE II 圖形遊戲系統設計

GAME! GAME! GAME!

1 登陸星球
金 樸 克
迷 宮
PIANO
螺 旋

暢銷書「第三波」作者托佛勒教授以「第三波」稱呼資訊時代，並以「波前(Wave front)」代表鼓動第三波風潮一些前進的事物。

無疑的，在台灣電腦與管理事業書籍的出版是第三波新文明的一「波前」之一，她將負起引導台灣社會邁向第三波的重大責任。

然而坊間電腦書籍，內容不盡理想，浪費讀者寶貴的時間與金錢。本公司有鑑於此，乃邀請專家學者有系統的撰寫一系列電腦與管理書籍，以最高品質。服務讀者，同時達到「波前出版，必屬好書」的口碑。

波前電腦管理圖書有限公司

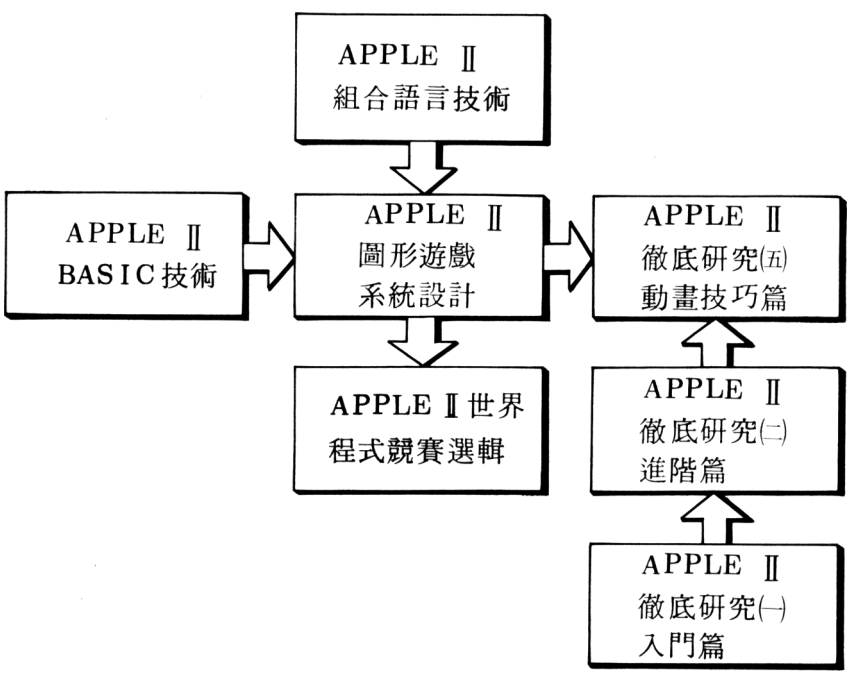
爲服務讀者，本書所有程式均製成磁片，請向波前公司直接郵購。

波前新書介紹

● Apple II 微電腦徹底研究(五)	施威銘	著
● Apple II 微電腦徹底研究(三)	施威銘	著
● Apple II 組合語言精華	李天佑	著
● Apple II 資料庫程式設計（即將出版）	施威銘	著
● Apple II 世界程式競賽選輯	林和賢	編著

波前建議您相關的參考書籍

提供您最有效率的電腦研習效果



序

許多人是因爲精彩有趣的GAME而進入微電腦海闊天空的領域。通常是由玩GAME開始，繼而對GAME程式的設計發生好奇，而終於造成不眠不休的研究狂熱。一般的情形是自己設計了一點簡單的程式，但終究覺得沒有市售成品的精彩好玩，速度上也相差甚多，好奇的人會拿一片小蜜蜂的磁片（磁帶）來觀摩其程式設計的技巧，往這方向走的，就開始一連串的解保護、造預移圖序、圖形壓縮等屬於機械語言的長期奮鬥了。

不過，相當多的人都因爲陌生的機械語言而無法更上層樓，進一步追求研習的樂趣。但是，使用 BASIC 一樣是可以造出優秀的遊戲來的！畢竟語言只是一種工具，如果因爲迷失於語言的困境中，而不能對程式的系統結構善加安排，說起來眞是本末倒置可惜的事情呢！對於初學者而言，經由易於瞭解的 BASIC語言來設計GAME程式，使得有足夠的精神來對程式結構策略加以研究，可說是最適當不過的事情了！

這本書便是以這樣的想法來編寫的，全書共分成兩大部份，前半部份是爲學習者提供技術上的研習與參考，其中第一章是將一般書籍較少提到的BASIC 繪圖技巧予以解說，在第二章當中，爲了解決相當耗時的圖形表設計，我們介紹了自動造表程式來簡化程式設計的工作，所以讀者可以很方便的使用DRAW 、

XDRAW 、ROT 等圖形而不必煩惱圖形表的設計。第 5、6 章則介紹了第 2 部份程式的架構，以及製造磁帶系統的詳細說明。在第二部份當中，我們共選輯了 10 個有趣的遊戲，這些程式均是經過了系統化的組合，你會發現除了故事內容的差異之外，整體上的構造及設計手法，都是有其一定的脈絡可尋的，希望經由對於這些程式的瞭解，能夠有助於你對於遊戲程式設計的學習。

總之，請開始閱讀這本書吧！相信你一定能在毫無困難的情況下，鍵入每一則程式而樂趣無窮的學習著！

林和賢

1983 07.20

台北．板橋

目錄

第一部份　技術講話

第 0 章　初學者的下一步…… 1

0 － 0　不知道問題在那裏！ 1

0 － 1　問題在那裏 2

0 － 2　本書結構 3

第 1 章　活用 APPLE 的圖形表…… 5

1 － 0　APPLE Ⅱ 的 3 種繪圖方式 5

1 － 1　APPLE 的圖形表 7

1 － 2　圖形表怎麼用？ 8

1 － 3　有關圖形表的特殊位址 9

1 － 4　頁次輪換的技巧 16

第 2 章　自動造表程式…… 19

2 － 0　圖形表爲什麼麻煩？ 19

2 － 1　磁碟機使用者的自動造表程式 23

2 － 2　磁帶使用者的自動造表程式 36

第 3 章　戲劇效果的文數字形圖形表…… 47

3 － 0　字形圖形表 47

3 － 1　字形圖形表的使用法 51

第 4 章　音響的效果…… 55
4 － 0　一個發聲的例子 56
4 － 1　另一個聲音程式 58

第二部份　程式輯

第 5 章　如何製造您的程式㈠…… 63
第 6 章　如何製造您的程式㈡…… 67
第 7 章　大蜘蛛…… 75
第 8 章　外星人登陸㈠…… 91
第 9 章　分秒不差……107
第10章　ＰＩＡＮＯ……129
第11章　迷宮……149
第12章　摘星星……165
第13章　螺旋……179
第14章　外星人登陸㈡……197
第15章　精打細算……217
第16章　金撲克……233
第17章　節目單……261
第18章　創造自己的 GAME ……269
參考書目……273

技術講話

APPLESOFT

的繪圖技巧詳解

第 0 章

初學者的下一步

0-0 不知道問題在那裏./

相信電腦的愛好者皆會同意，無論您學了電腦一星期、一個月甚或一年，總會有那麼一個時候，當您面對著美麗而精緻的電腦畫面，不禁要問：這樣的程式是如何設計出來的呢？有哪些書本或資料可以教我有關的知識呢？而我是否也可以來設計這樣的程式？

於是，您便嘗試寫了一些程式，結果當然也有一番體會……

問題的癥結——我們的問題是因爲不知道問題在哪裏

那麼，問題在哪裡？我們面對的困難是什麼呢？說起來，這也是電腦快速流行之後而普遍引起的一個現象：當學完 PRINT、FOR…NEXT、GOSUB 、……之後的下一步，爲什麼竟然是舉筆維艱的寫不出一個有用的程式呢！

包括作者在內的非科班出身的電腦人都曾經有這樣的經歷吧！那就是：我們所面對的問題是因爲不知道問題在哪裡！

0-1 問題在那裏？

對一個沒有經過正規訓練（？）的電腦愛好者來說，設計程式就好像沒有經過寫作訓練的人寫文章一般的，只憑靈感而不考慮格式、結構、文體吧？誠然，靈感可能寫出動人的詩篇，而經世的論述卻必須眞實的才學了！

所以，這便是問題的所在之一：我們必須學習如何規劃、安排、有系統的來設計程式，一旦程式設計有了一定的規範、有了可循的程序，那種不知如何着手或寫不出較大程式的困難，便迎刃而解了，這種結構化的程式設計方法（不只程式本身結構化，程式設計過程也結構化），尤以本身最不結構化的BASIC*及組合語言** 最爲需要。

另外一個程式設計上的問題是，系統熟悉運用的程度，譬如，一直令初學者既好奇又困惑的PEEK、POKE指令便是一個例子，其實PEEK、POKE並不是什麼難以瞭解的指令，只不過是因爲這些指令可以直接的改變系統狀態，而造出奇特的結果，如此一來，便使得對電腦系統構造不熟悉的初學者感到迷惑了。

明白了問題之後，我們要做的便是：熟悉 APPLE 繪圖方面的特殊功能，並加以結構化的運用。

---

* 結構化BASIC　関國田譯　波前電腦圖書公司

** APPLEⅡ 徹底研究(3)動畫技巧篇　施威銘著　波前電腦圖書公司

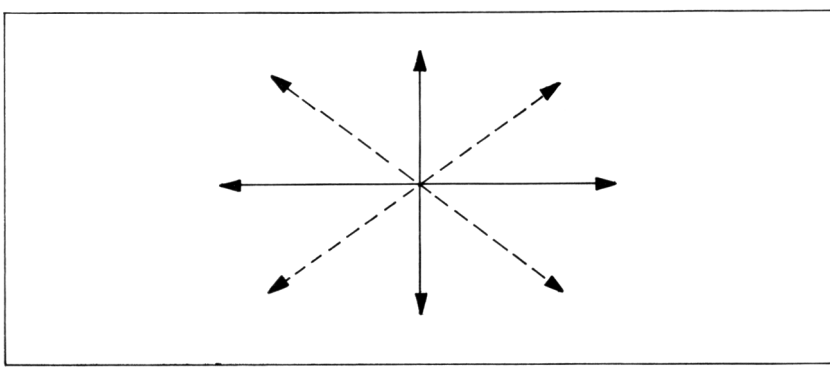
0-2 本書結構

現在，我們可以來介紹一下本書的內容了。本書共分爲兩大部份，前半部份是爲後半部份的程式設計而做的技術提要，其主要目的是在對於已熟悉 Applesoft 的讀者提供一些建議，譬如說，大家都知道 Applesoft 的圖形表繪圖指令，但使用起來却覺得很困難。到底要如何有效的使用圖形表繪圖呢？便是本書要討論的一個問題了。另外，對於磁碟機的使用者，可以在程式執行當中存取磁片資料，是一件很方便的事情，可是磁帶錄音機的使用者呢？本書也介紹您一種有趣的磁帶程式製造法，你也可以造出有保護的商品磁帶呢！

至於後半部份的程式，則是一系列有趣而系統化設計的遊戲，全部程式均是以前述的圖形表方法來造出吸引人的畫面動作，你可以經由這些程式而體會到向量繪圖法（圖形表法）的極限發揮。不過，除了遊戲的趣味性之外，我們更希望您能夠瞭解，一個程式的結構化與否，是如何的影響到其是否爲一個優秀的作品呢！

第 1 章

活用APPLE的圖形表



1-0 APPLE II的3種繪圖方式

基本上APPLE 的繪圖方式有三種，各種方法均有其適用的場合及優缺點，現在分述如下：

1.向量繪圖法(XDRAW與DRAW)

這是大家熟悉的一種方式*，只要設計者繪出圖形，依規則造好圖形表，然後使用Applesoft 內含的繪圖指令XDRAW 及DRAW便可以將圖形畫在螢幕上。這種

* 參考資料Ref . 1

方法的好處是不需要額外的繪圖副程式，只要將圖形表附加到程式內，以BASIC 便可以畫出圖形來，並且使用 ROT 及SCALE 更可以令圖形旋轉，放大。其中DRAW 是單純的繪圖指令，而 XDRAW 則是具有EOR(斥或邏輯）特性的繪圖指令，因此如果第一次 XDRAW 是用來畫出圖形的話，則第二次在原位置上使用 XDRAW ，會將第一次所畫的圖形消去，但保留了背景。

向量繪圖法的缺點是：①只能畫單色圖形，也就是說，在一個圖形內部只有一種顏色。②不易畫實心圖形，這是向量繪圖的先天限制。③速度較慢，尤其是畫較大圖形時，更形明顯。④圖形表設計、換算，費時費力。

所以，通常向量繪圖法是使用在較小圖形、較少圖形、速度要求不嚴格的場合，不過，巧妙的運用，却也一樣能夠造出優秀而吸引人的效果來，本書的程式便是幾乎全部以這種方法來設計的。至於圖形表設計、換算的問題，我們會令Apple 自動的來爲我們處理。

2.HPLOT副程式繪圖法

這是直接利用Apple內部的HPLOT 及HPLOT TO 等指令所屬的ROM 副程式來繪圖的一種方法。這種方法，由於使用了機械語言程式，因此速度上比較快（因爲不必經過DRAW、ROT、SCALE 等副程式處理），並且對於圖形上的多重彩色、實心圖形等問題，也可以獲得合理的解決。

3.位元映射繪圖法

雖然HPLOT 方式的繪圖速度已獲改善，但是由於HPLOT 是一個考慮到多方面適用性的副程式，因此必然會相互遷就而使得執行的效率（速度）有所折扣。位元映射繪圖法，是直接將圖形存入到畫面的記憶位址內，而立即表現到畫面上的繪圖方法。位元映射繪圖的方式也分成好幾種，在APPLE Ⅱ微電腦微底研究第 5 册之中曾經介紹了一種利用畫面位址查表法及預移圖序法所構成的高效率繪圖法，有興趣讀者可參考之。這種位元映射繪圖法幾乎是大多數現代專業動畫程式所採用的方法了。

1-1 APPLE的圖形表

前述的 3 種繪圖法，其中以向量繪圖的圖形表法最適合在BASIC 程式中使用，這是因爲Applesoft 當中已含有ROT、SCALE、XDRAW、DRAW等指令，只要將圖形表造出來便很方便使用了。既然本書是以BASIC 爲主要語言，所以我們便來探討一下如何活用Apple的圖形表。

許多初學Apple的人，一開始都會因爲圖形表指令具有旋轉及放大的繪圖功能而甚感興趣，可是嘗試了幾次之後，却又多因圖形表本身繁複的換算手續而放棄了。其實，最初幾次動手來做換算的工作是必須的，由此，您才可以體會出其基本規則，但是眞正的實用設計，却大可不必這樣一步一點的來加以計算，這種

工作，我們將在第二章介紹一個自動造表程式來爲我們代勞。

所以，請參照Apple原始說明，依所舉的例子畫個圖形看看，此後，您就不必再做這種工作了！因爲，這只不過是Apple語言系統的一種規則，對電腦技術的學習，並無多大幫助！

1-2 圖形表怎麼用？

只要有了自動造表程式，可以自動爲我們換算圖形表之後，我們就可以不要問〝圖形表怎樣做〞，而是問〝圖形表怎麼用〞了！

使用圖形表，首先要考慮的是圖形表存放的位址，如果是太長的表格（如第3章的字形表），則不適於存放在第3頁甚或是高解像畫面之前（小於＄2000）的位址，因爲這樣可能影響BASIC 程式的使用空間，另外一些陳列變數的存放區也要特別注意勿與之重疊。在本書當中，我們選擇高解像第2頁畫面之後的區域（大於＄5FFF）來存放圖形表，字形圖形表是由＄61A8（25 000）開始存放，而圖案圖形表則由＄6978（27 000）開始。

兩組以上的圖形表

通常，圖形表是只有一組的，但是像本書這樣系統化的一次設計出多個程式的場合，總是有如文數字形這種幾乎每則程式均可共同使用的圖形表。爲了令這種〝公用的〞圖形表方便於附加到各個程式上，我們就必須將它自成一組的與程式個別的圖案圖

形分開來。這便是我們將字形圖形表置於$ 61A8 而圖案圖形表則存放於$ 6978 的原因了。至於爲什麼採用這兩個奇怪的位址做圖形表起始點呢？因爲它們的十進位值分別是25000 及27000 ，在BASIC 程式中的計算較方便。

但是有了兩組不同的圖形表又怎麼辦呢？

1-3 有關圖形表的特殊位址

要使用Apple的圖形表指令繪圖，除了將圖形表造好、存入記憶體之外，還要告訴Apple圖形表的起始位址才行。Apple的圖形表起始位址指位器是（$E8 ，$E9）＝（232,233），所以，如果你要畫出英文字母，就必須以告訴Apple說我現在要

```
POKE 232,168
POKE 233,97
```

DRAW（或XDRAW ）的圖形是在由$ 61A8 開始的這組圖形表當中。其中232所存的是$61A8 的低位元組（$A8 = 168），233所存的是$ 61A8 的高位元組（ $ 61＝97 ）。

碰撞旗號

除了（ $E8,$E9 ）這組位址之外，Apple還有一個與DRAW,XDRAW有關的位址： $EA＝234。這個緊接在（ $E8，$E9）的位址，有一個很有趣的功能，那就是每次在執行DRAW指令時，如果發現所欲畫點（位元）的位置已經畫了點（該位置所對應的位元值＝1 ）或所欲消點的點已被消去（該位置所對應

的位元值＝0），則$EA的內含值便加一，因此$EA內含值的增加值便是碰撞次數了。譬如一艘白色的太空船在黑色的背景上移動，事實上便是將黑色的背景畫上白色的圖點（將bit＝0變成bit＝1）。這時如果發現所欲畫點的位置其位元值爲1，則必然是碰上黑色背景以外的物體了（因爲若是黑色背景，位元值爲0而不是1）。

爲了說明碰撞旗號的功能，此處特別舉了一個自動塗色的程式，您可以將之與APPLE II徹底研究第五册所提到的另一個自動塗色程式比較一下其優缺點。

自動塗色程式

```
]LIST

10  REM   ****************
20  REM   * COLOR-FILLER *
30  REM   ****************
40  REM
50  REM
100  REM  ==============
110  REM     INITIALIZE
120  REM  ==============
130  HGR : SCALE= 1: ROT= 0: HCOLOR= 3
140 MOVEREG =  - 2613
150 HY = 226
160 CLR = 62450
170 TABLE = 768: REM  THIS VALUE RELOCA
     TES EVERYTHING
```

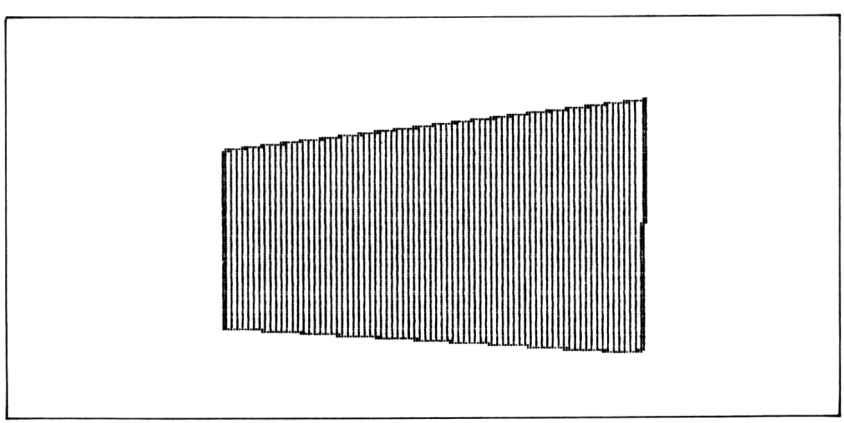
```
180 C$ = "GREEN"
200  REM  ===========
210  REM      POKES
220  REM  ===========
230 SHAPE = TABLE + 4:FILL = TABLE + 6
240  FOR SPOT = (TABLE) TO TABLE + 19: READ
     CODE: POKE SPOT,CODE: NEXT
250  DATA  1,0,4,0,4,0,162,4,160,3,32,1
     ,246,169,0,197,234,240,243,96
260 MSB =  INT (TABLE / 256):LSB = TABL
     E - 256 * MSB
270  POKE 232,LSB: POKE 233,MSB
300 MSB =  INT (SHAPE / 256):LSB = SHAP
     E - 256 * MSB
310  POKE FILL + 1,LSB: POKE FILL + 3,M
     SB
320  REM  ===============
321  REM    HPLOT   BOX
322  REM  ===============
330  HOME : CALL CLR
340 X(1) =  RND (1) * 100:X(4) =  RND (
     1) * 100
350 X(2) =  RND (1) * 100 + 170:X(3) =
      RND (1) * 100 + 170
360 Y(1) =  RND (1) * 70:Y(2) =  RND (1
     ) * 70
370 Y(3) =  RND (1) * 70 + 90:Y(4) =  RND
     (1) * 70 + 90
380  HPLOT X(1),Y(1) TO X(2),Y(2) TO X(
     3),Y(3) TO X(4),Y(4) TO X(1),Y(1)
390  HPLOT X(1) + 1,Y(1) TO X(2) + 1,Y(
     2) TO X(3) + 1,Y(3) TO X(4) + 1,Y(
     4) TO X(1) + 1,Y(1)
400 XO = (X(1) + X(2) + X(3) + X(4)) /
     4:YO = (Y(1) + Y(2) + Y(3) + Y(4))
      / 4
```

```
410 XO = 2 * ( INT (XO / 2))
420  IF C$ = "GREEN" OR C$ = "ORANGE" THEN
     XO = XO + 1
430  REM  ====================
431  REM     FILL FIGURES!!!
432  REM  ====================
440 X = XO:INC = 2
450 Y = YO
460  POKE SHAPE,4: GOSUB 690
470  IF  ABS (CY - Y) < 2 THEN 530
480 TY = CY
490  POKE SHAPE,6: GOSUB 690
500  IF  ABS (CY - Y) < 3 THEN 530
510 BY = CY:Y =  INT (TY + BY) / 2
520 X = X + INC: IF X < 280 AND X > 0 THEN
     460
530  IF INC > 0 THEN INC =  - INC:X = X
     0 + INC: GOTO 450
540  IF  ABS (CY - Y) < 3 THEN 560
550 X = X - 1: IF X > 0 THEN 490
560  REM  ================
561  REM     NEXT FIGURE
562  REM  ================
570  IF C$ = "GREEN" THEN C$ = "VIOLET"
     : GOTO 610
580  IF C$ = "VIOLET" THEN C$ = "ORANGE
     ": GOTO 610
590  IF C$ = "ORANGE" THEN C$ = "BLUE":
      GOTO 610
600  IF C$ = "BLUE" THEN C$ = "GREEN"
610  HCOLOR= 3: IF C$ = "BLUE" OR C$ =
     "ORANGE" THEN  HCOLOR= 7
620  FOR PAUSE = 1 TO 100
630  IF  PEEK ( - 16384) < 127 THEN 670

640  GET G$
```

```
660  IF G$ =  CHR$ (27) THEN  HOME : TEXT
     : END
670  NEXT
680  GOTO 320
689  REM ==================
690  REM  PAINTING ROUTINE
691  REM ==================
700  DRAW 1 AT X,Y
710  CALL FILL
720  CALL MOVEREG
730 CY =  PEEK (HY)
740  RETURN
```

程式1－0　自動塗色程式

這個程式執行時，會在高解像畫面以RND函數畫出任意四邊形，然後將四邊形的內部塗色，如果您按下ESC鍵，則塗完該四邊形後便結束了，否則程式以一個迴圈不斷的畫出任意四邊形及塗色。



塗色程式的執行例

自動塗色程式說明

這個程式基本上是利用$EA這個碰撞旗號來做塗色的判斷，塗色動作由圖形的中央開始向上、下、左、右以DRAW來畫出一點（因爲要用$EA旗號，因此不用HPLOT），當發現碰撞（$EA的內含值不爲0）則停止畫圖，並且改換畫圖方向。基本上這些動作是由存改在$306的機械語言副程式來完成的：

```
*306L

0306- A2 04     LDX   #$04    ; 指向圖形資料
0308- A0 03     LDY   #$03    ; 眞正的起始位址
030A- 20 01 F6  JSR   $F601   ; DRAW
030D- A9 00     LDA   #$00
030F- C5 EA     CMP   $EA     ; ($EA) = 0 ? (不碰撞?)
0311- F0 F3     BEQ   $0306   ; 是，至$306再畫圖
0313- 60        RTS           ; 碰到邊界，則RTS
```

程式1－1　塗色副程式

程式1—1當中，$F601 就是DRAW的進入點，使用$F601 這個副程式時，X—暫存器要存入圖形表眞正起始位址的底位元組，Y—暫存器則要存入其高位元組，在我們的例子裡，圖形表是由$0300 開始，但其4位元組爲圖形個數與距離位元組，因此頁址的圖形表是由$0304 開始，所以程式1—1當中X是存入#$04，Y是存入#$03。

程式1—0由100～180行設定變數初值，其中MOVEREG=—2613是一個ROM的進入點，這個副程式將所畫圖點的指標取入零頁的224，225（存X指標）及226（Y指標）位址內，由

此我們可以知道畫圖的位置。CLR = 62450 則是將高解像畫面清除的副程式。TABLE = 768 是圖形表的開始位置。 C$ 是目前所塗的顏色。

由 200～310行是將圖形表存入 300～305 共6個位址內，並且也把程式 1 —1 存入到 $306～$313 位址，其中 260～270行是將圖形表起始位址存入前述的（232，233）圖形表起始位址指位器內。300～310 行則是將眞正的圖形資料位址送入到程式 1 — 0 當中，因爲 FILL+1（=$307）及 FILL+3（=$309），應該存入的值正是 #$04 及 #$03 。320～420 行以亂數畫出任意四邊形，設（X0，Y0）爲其中點， C$ 爲其顏色。430～680則做塗色換色的工作，700 ～ 740行才是眞正做塗色動作的程式，其中 FILL（=$306） 副程式我們已提過，就是繪圖副程式。但是，爲什麼在CALL FILL 之前還有一個 DRAW 1 AT X，Y呢？

這是因爲 $306 的副程式並未指定畫圖的位置，其中 $F601 是DRAW 副程式的一部份，它的功能是在上一次繪圖的下一點位置上畫圖，向量繪圖是有方向性的，因此有下一點，所以我們就必須先以 DRAW 在四邊形的中央（由程式第 400 行取平均值）畫出一點，然後 $306 副程式中的 $F601 才可以依其方向來繼續畫點，由於第 460 行的 POKE SHAPE，4 是代表〝↑〞向上繪圖的向量（Apple 圖形表規則），因此 DRAW 之後的下一點便是再往上的一點，而第 490 行的 POKE SHAPE，6 是代表〝↓〞向下繪圖的向量，因此下一點是往下的。

$306 的副程式畫出第一點之後，便檢查碰撞旗號 $EA 的內含值，若內含值爲 0 ，則表示尙未到達邊界，所以繼續往此方向

畫點，若$EA的內含值不爲0，則表示發生碰撞（即到達邊界），於是便停止畫圖而回到主程式了。執行完460行及490行之後，便已向上、下畫完一條垂直線了，接著向右畫完半個方塊再向左完成整個工作。至於720行的MOVEREG（=-2613）副程式，則是用來將畫圖方向下一點的Y座標找出，並存於$E2（=226）的副程式。此處，我們用它來找出四邊形上下邊界的Y座標值，以便在第510行取平均值以做爲下一行畫圖的中點（請記住，我們是由中點往上下畫線的），所以請執行一下我們的程式，看看是否塗色動作中的畫圖中點是隨著四邊形的形狀而作動態調整呢？

1-4 頁次輪換的技巧

APPLE 無論是在高低解像畫面或是文字幕，均有雙頁的畫面位址，經由軟體開關（$C054,$C055）可以分別令第一頁或第二頁的內容顯現到螢幕上來，由於軟體開關的轉換只須幾個微秒（μS）的時間，因此其畫面速度自然是遠非任何畫圖動作所可比擬的了。這種技巧常被用於高級的動畫程式當中，底下的程式便是一個有趣的例子：

```
]LIST

10  HGR : HCOLOR= 3: HPLOT 0,0 TO 279
    ,191: HPLOT 279,0 TO 0,191: HGR2
    : HPLOT 140,0 TO 140,191
20  FOR I = 768 TO 795: READ WW: POKE
    I,WW: NEXT I: CALL 768
100  DATA  173,80,192,173,87,192
110  DATA 173,84,192,169,11,32
120  DATA 168,252,173,85,192,169
130  DATA 11,32,168,252,173,0
140  DATA 192,16,235,96
```

程式1－2　頁次輪替程式

```
*300L

0300-   AD 50 C0    LDA   $C050
0303-   AD 57 C0    LDA   $C057
0306-   AD 54 C0    LDA   $C054
0309-   A9 0B       LDA   #$0B
030B-   20 A8 FC    JSR   $FCA8
030E-   AD 55 C0    LDA   $C055
0311-   A9 0B       LDA   #$0B
0313-   20 A8 FC    JSR   $FCA8
0316-   AD 00 C0    LDA   $C000
0319-   10 EB       BPL   $0306
031B-   60          RTS
```

程式1－3　頁次輪替副程式

程式 1 — 3 是程式 1 — 2 第 20 行所 POKE 進來的機械語言，它的作用就是在板動 $C054 及 $C055 這兩個軟體開關，使畫面呈現第一頁與第二頁交替的動作，由於很快速的換頁，於是你便看到有趣的視覺效果了。

雖然這個程式是將兩頁上已畫好的圖形替換出現，但是爲什麼看起來有連續動作的效果呢？說起來這是頁次輪替法的一個難題，因爲看起來靜止的電視畫面，其實是不停地在做掃瞄動作，由於換頁動作的速度較電視掃瞄動作快上許多，因此可能第一頁的畫面掃瞄到一半就換過第二頁，而第二頁的內容又是從半途接下掃瞄，並且由什麼地方開始接替，也不是每幅圖場均是一致，所以便發生畫面變動的情形了，這種情形就是所謂視覺差頻同步的效果。雖然這是一個不易掌握的問題，但是適當的使用，反而可以造出像程式 1 — 2 的有趣畫面呢！在第 17 章的 MENU 也使用了這個技巧，請參考之。

第 2 章

自動造表程式

APPLE 的圖形表繪圖法雖然是很實用，但造起圖形來却是不勝其煩，因此本章介紹您很實用的自動造表程式，無論你是磁片或磁帶用者，均可以很方便的來使用。但是，如果你並不正在設計自己的 GAME，而且想快點嘗試第 2 部份的遊戲的話，則請跳過本章……

2-0 圖形表爲什麼麻煩？

APPLE 的圖形表是一種較爲低階的語言（造表規則），這是爲了適應其XDRAW、ROT等指令之使用格式而作的一種表格規則，爲了適應指令而造的低階格式，當然就無法成爲方便吾人使用的工具了，大概這便是APPLE 圖形表不方便使用的原因吧？

那麼要怎樣才可能造出方便吾人使用的圖形表呢？方法很多，在於是否巧妙的構思而已。譬如，您可以創造出一套簡便使用的圖形表格式，以“SDATA”(SHAPE DATA) 來代表圖形資料

，“SDATA”的內容除了索引之外便只有U(上)、D(下)、L(左)、R(右)，並且以“SREAD”(SHAPE READ)來讀取這些U、D、L、R，這樣子，就可省去麻煩的換算手續了！至於怎樣創造“SREAD”、“SDATA”呢*？就請您來嘗試吧！

畫圖向量	代碼	
↑	000	只做移動不畫點
→	001 或 01	
↓	010 或 10	
←	011 或 11	
↑	100	先畫點再移動
→	101	
↓	110	
←	111	

圖 2—0

圖 2—0 當中的代碼是由位元所構成的，一個位元組的 8 個位元是被分割成如下的狀況：

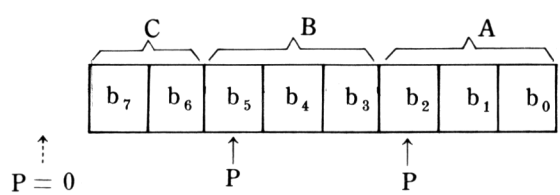
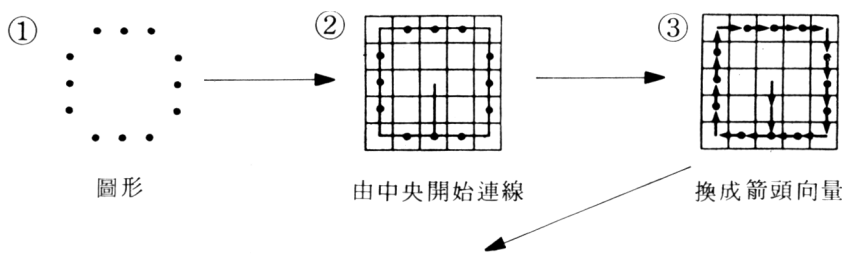


圖 2—1

* Ref.6 第 3 章

其中A、B是3個位元一組的碼，其最高位元（A的b_2及B的b_5）由圖2—0當中可以知道是用來控制是否畫點的位元，而其它的兩個位元則用來指定移動的方向。至於C，則只有兩個位元，可以看成最高位元的P是0，這樣的話，只適合表示圖2—0當中不畫點的繪圖向量，而且↑這個向量也不可以由C部份來設定，因爲↑必須眞正用3個0來代表，而C只有兩個位元而已，這是我們設計自動造表程式時特別要注意的地方。

所以，像下圖這樣的方形，其向量及代碼便可以依圖2—0來造出：



補0及換算成位元碼

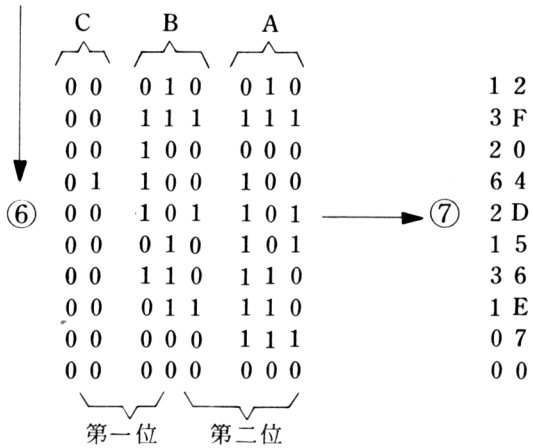
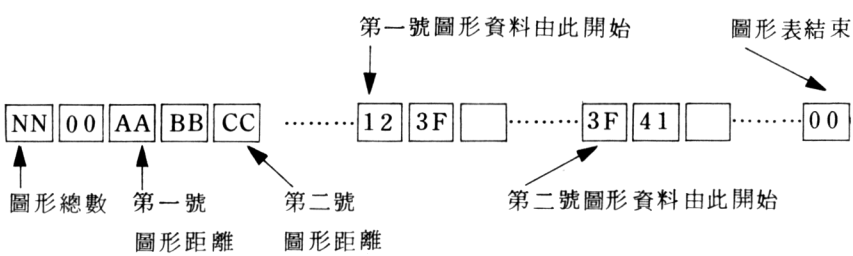


圖 2—2

經由上列①～⑦步驟，我們終於造出一個方形的圖形表了。但是工作尚未結束，由於APPLE 的圖形表可以存放一個以上的圖形，因此我們必須在圖形表的開頭加上索引表格，以告訴APPLE 圖形表共有多少圖形、哪一號圖形由哪裡開始？索引的格式如下圖示：



由此，你可以體會出造一個圖形表是怎樣繁複的事情了。

2-1 磁碟機使用者的自動造表程式

上一節介紹的圖形表製作，可說是件既費時費事而又無任何意義的工作，因爲它畢竟只是一項硬性的規矩，而無法對技術的學習提供任何益處。因此，只要我們能夠設計一個自動製造圖形表的程式，那麼一切煩雜的工作，便可以交由電腦處理而不必勞我們費神了，底下便是一個這樣的程式：

```
10  REM  ************************
20  REM  **    SHAPE   MAKER    **
30  REM  ************************
55  CLEAR
60  DIM VV$(8),VS%(505),SR$(8)
70  DIM DQ(8),RS%(10),BS$(8)
80  REM  ====POKE COUSOR====
81  ROT= 0: SCALE= 1
82  FOR I = 0 TO 5: READ WW
84  POKE 768 + I,WW: NEXT I
85  DATA   1,0,4,0,4,0
90  POKE 232,0: POKE 233,3
100  REM ==== MAIN ====
105  GOSUB 2100: GOSUB 2000
110  GOSUB 2050: GOTO 200
200  REM  -- GET KEY --
205  HTAB 20: VTAB 24:XX =  PEEK ( - 16384)
206  IF XX < 128 THEN  XDRAW 1 AT X,Y: XDRAW
     1 AT X,Y: GOTO 205
207 A$ =  CHR$ (XX - 128): POKE  - 16368,0
208  PRINT A$;
210  IF A$ = "R" THEN  GOSUB 400
215  IF A$ = "E" THEN  GOSUB 500
```

```
220  IF A$ = "S" THEN  GOSUB 600
225  IF A$ = "G" THEN  GOSUB 1200
230  IF A$ = "#" THEN  GOTO 10
240  IF A$ = "!" THEN  TEXT : HOME :
     END
249  GOSUB 300: GOTO 205
250  REM  -- UP VECTORS? --
255  IF A = 2 THEN SW = SW + 1
260  IF SW = 3 THEN SW = SW - 2: GOTO
     285
270  IF SW = 2 AND A > 4 THEN SW = 0
     : GOTO 285
275  IF A = 2 THEN  RETURN
280 SW = 0: RETURN
285 VS%(N) = 1:N = N + 1:VS%(N) = 3:
    N = N + 1:VS%(N) = A
299  RETURN
300  REM  --- VECTOR PLOT ---
305 A =  VAL (A$)
308  IF A < 1 OR A > 8 THEN  RETURN

310  IF N + 1 > VL THEN MS = 4: GOSUB
     1500: RETURN
312 N = N + 1:VS%(N) = A
314  GOSUB 250
320  HTAB 35: VTAB 21: PRINT N;
325  IF A > 4 THEN  HPLOT X,Y
330 X = X + X(A):Y = Y + Y(A)
335  RETURN
350  REM  --- VECTOR ERASE ---
355 A = VS%(N)
360 X = X + ( - X(A)):Y = Y + ( - Y(
    A))
365  HPLOT X,Y
385 N = N - 1: HTAB 35: VTAB 21: PRINT
    N;
```

```
399  RETURN
400  REM  -- REPEAT --
404 AH$ = A$
406  HTAB 1: VTAB 22
410  INVERSE : PRINT SR$(3);: NORMAL

412 HB = 8:CH = 3: HTAB 7: VTAB 22: INPUT
    IN$
414 RV =  VAL (IN$)
416  INVERSE :HB = 20:CH = 10: HTAB
    10: VTAB 22: INPUT "VECTOR?";IN$
    : NORMAL
418  GOSUB 950: IF NF = 1 THEN  RETURN

420  REM  -GET REPEAT VECTORS-
422 Z =  LEN (IN$)
425  FOR I = 1 TO Z
430 RS%(I) =  VAL ( MID$ (IN$,I,1))
435  NEXT
449  REM  -- VALID VECTOR --
450  FOR I = 1 TO Z
452  IF RS%(I) > 0 AND RS%(I) < 9 THEN
     NEXT I: GOTO 460
454 MS = 2: GOSUB 1500:I = Z
456  NEXT I: GOTO 499
459  REM  -- PLOT REPEAT --
460  FOR H = 1 TO RV
462  FOR J = 1 TO Z
464 A$ =  STR$ (RS%(J)): GOSUB 300
466  NEXT J: NEXT H:Z = 0
499 A$ = AH$: RETURN
500  REM  -- ERASE --
505  HTAB 1: VTAB 22
510  INVERSE : PRINT SR$(2);: NORMAL

515 HB = 8:CH = 3: HTAB 7: INPUT IN$
```

```
520  IF NF = 1 THEN  RETURN
525 EE =  VAL (IN$)
530  HCOLOR= 0
560  IF EE = 0 THEN EE = 1
565  FOR I = 1 TO EE
570  IF N = 0 THEN I = EE: GOTO 590
585  GOSUB 350
590  NEXT I
595  HTAB 35: VTAB 21: PRINT "   ";
596  HTAB 35: VTAB 21: PRINT N;
599  HCOLOR= 3: RETURN
600  REM  --- SAVE SHAPE ---
602  IF N = 0 THEN  RETURN
604 C =  FRE (0)
605  HTAB 1: VTAB 22
610  INVERSE : PRINT SR$(1);: NORMAL

615  HTAB 6: VTAB 22: INPUT IN$
620  GOSUB 950: IF NF = 1 THEN  RETURN

625 NA$ = IN$
628 AV = 27000:VA = AV
630  HTAB 28: VTAB 22: FLASH : PRINT
    "PLEASE WAIT";: NORMAL
635 AA = 0:BB = 0:CC = 0
640  FOR I = 1 TO N
650 J = VS%(I): GOSUB 700
660  NEXT
682  IF AA ≠ 0 THEN  GOTO 689
684  IF BB = 0 THEN B$ = "000"
686 C$ = "00"
688  GOSUB 800
689  POKE AV,0:AV = AV + 1
690  GOSUB 900
695  RETURN
700  REM  -COMBINE VECTORS-
```

```
705  IF AA = 1 THEN  GOTO 720
710 A$ = VV$(J):AA = 1: GOTO 799
715 PA = J
720  IF BB = 1 THEN  GOTO 740
725 PB = J
730 B$ = VV$(J):BB = 1: GOTO 799
740 C$ =  MID$ (VV$(J),2,2)
750  IF J = 2 OR J > 4 THEN C$ = "00
    ":I = I - 1
760  IF PB = 2 AND (J = 2 OR J > 4) THEN
    B$ = "000":I = I - 1
780  GOSUB 800
799  RETURN
800  REM  ---- BINARY CONV ----
810 D$ = C$ + B$ + A$
820  FOR IQ = 1 TO 8
830 DQ(IQ) =  VAL ( MID$ (D$,IQ,1))
835  NEXT IQ
840 AD = DQ(1) * 128 + DQ(2) * 64 +
    DQ(3) * 32 + DQ(4) * 16 + DQ(5) *
    8 + DQ(6) * 4 + DQ(7) * 2 + DQ(8
    )
842  POKE AV,AD:AV = AV + 1
845 AA = 0:BB = 0:CC = 0
848  RETURN
900  REM  -- BSAVE SHAPE --
905 LN = AV - VA
910  PRINT CC$;"BSAVE ";NA$;",A";VA;
    ",L";LN
915  HTAB 28: VTAB 22: PRINT "
         "
920  HTAB 35: VTAB 22: PRINT LN;
949  RETURN
950  REM  -- NAME CHECK --
955 NF = 0: IF IN$ = "" THEN NF = 1:
     RETURN
```

```
960  IF  ASC (IN$) < 48 OR  ASC (IN$
    ) > 90 THEN NF = 1
970  RETURN
999  RETURN
1200  REM  -- GET SHAPE --
1222 C =  FRE (0)
1234  HTAB 1: VTAB 22: INVERSE
1236  PRINT SR$(1);: NORMAL
1238  HTAB 6: VTAB 22: INPUT IN$
1240  GOSUB 950: IF NF = 1 THEN  RETURN

1242 X = 140:Y = 80:NN = 1:N = 0
1244  HGR :NA$ = IN$
1245  ONERR  GOTO 1399
1246  PRINT CC$;"BLOAD ";NA$
1248  HTAB 35: VTAB 21: PRINT "    ";

1250 VA =  PEEK ( - 21902) +  PEEK (
    - 21901) * 256
1255 LN =  PEEK ( - 21920) +  PEEK (
    - 21919) * 256
1260  HTAB 28: VTAB 22: FLASH : PRINT
   "PLEASE WAIT";: NORMAL
1280  REM  -CONVERT VECTOR-
1281 VA = VA - 1
1282  FOR I = 1 TO LN - 1
1284 DV =  PEEK (VA + I)
1286  FOR J = 1 TO 8
1287 BS$(J) =  STR$ (0)
1288  IF DV >  = DC THEN BS$(J) =  STR$
    (1):DV = DV - DC
1290 DC = DC / 2: NEXT J
1292 C$ = "0" + BS$(1) + BS$(2)
1294 B$ = BS$(3) + BS$(4) + BS$(5)
1296 A$ = BS$(6) + BS$(7) + BS$(8)
1300 S$ = A$: GOSUB 1360
```

```
1310  IF B$ = "000" AND C$ = "000" THEN
     GOTO 1348
1320 S$ = B$: GOSUB 1360
1330  IF C$ = "000" THEN  GOTO 1348
1340 S$ = C$: GOSUB 1360
1348 DC = 128
1350  NEXT I
1352  REM
1354  REM  -- PLOT VECTORS --
1355  FOR J = 1 TO NN - 1
1356 A = VS%(J): GOSUB 310
1358  NEXT : GOTO 1380
1359  REM  -- STORE VECTORS --
1360  FOR J = 1 TO 8
1365  IF VV$(J) = S$ THEN VS%(NN) =
    J
1370  NEXT J:NN = NN + 1: RETURN
1380  HTAB 28: VTAB 22: PRINT "
           ";
1390  HTAB 28: VTAB 22: PRINT VA + 1
    ;"  ";LN;
1395  RETURN
1399 MS = 3: GOSUB 1500: POKE 216,0:
     GOTO 1200
1500  REM  -- ERROR --
1510  HTAB 1: VTAB 23
1520  PRINT SR$(4 + MS);
1530  GET E$
1540  IF  ASC (E$) = 13 THEN 1530
1545 H = RV:J = Z
1550  HTAB 1: VTAB 23: PRINT SR$(5);
1599  RETURN
2000  REM  -- INIT --
2004 AR = 17000:VL = 500
2010  HCOLOR= 3:NM = 1:UC = 0:LM = 5
    00
```

```
2030 CC$ =  CHR$ (13) +  CHR$ (4)
2040 X = 140:Y = 80:DC = 128
2045  HOME : HGR : SCALE= 1
2049  RETURN
2050  REM  -- SETUP
2055  HTAB 1: VTAB 21: PRINT SR$(4);

2060  HTAB 1: VTAB 23: PRINT SR$(5);

2070  RETURN
2100  REM  --- DATA ---
2102  DATA  -1,0,0,-1,1,0,0,1
2104  DATA  -1,0,0,-1,1,0,0,1
2110  DATA  "NAME
                "
2120  DATA  "NUMBER
                 "
2124  DATA  "NUMBER       VECTOR
                          "
2126  DATA "SAVE      REPEAT          #
    CLEAR"
2128  DATA   "GET       ERASE
     !END      1-8/PLOT"
2132  DATA   "INVAID VECTOR
                         "
2134  DATA  "FILE NOT FOUND
                          "
2136  DATA  "VECTOR LIMIT REACHED
                         "
2140  DATA  "011","000","001","010"
2142  DATA  "111","100","101","110"
2148  REM  --- READ DATA ---
2150  FOR I = 1 TO 8: READ X(I),Y(I)
    : NEXT
2170  FOR I = 1 TO 8: READ SR$(I): NEXT
```

```
2180  FOR I = 1 TO 8: READ VV$(I): NEXT

2199  RETURN

]
```

程式　2—0　使用磁碟機的自動造表程式

程式2—0的基本結構是700行～848行的向量組合（COMBINE VECTOR）及向量換算（BINARY CONV），這些工作是依照圖2—0、2—1、2—2的規則來做的，在圖2—0當中↑向量的值較爲特殊，這是因爲C部份只能當移動向量不能當畫圖向量，而如果在B之後仍需畫圖，則C必須略過（因C不能畫圖），我們是以將C塡入“00”來略過C部份，如果↑的碼再用“00”的話，則勢必以略過的情形相混淆，因此↑與其它3個向量不同，必須用“000”而不能用“00”，在程式2—0當中，這個不同點是由250行～299行來加以處理的。

程式2-0的操作法

1～8：分別作下列的向量畫圖工作

1＝	左移	5＝	畫點左移
2＝	上移	6＝	畫點上移
3＝	右移	7＝	畫點右移
4＝	下移	8＝	畫點下移

E ：鍵入E，字幕上會訊問NUMBER ？這時若鍵入一個數，譬如3，則便可以將前三次的繪圖向量從畫面上消去。

R ：鍵入R，字幕上會訊問NUMBER ？及VECTOR ？譬如你回答10及56，則畫面上將會把5、6所代表的向量（向上、下繪圖）重復畫10次（ 45° 的粗直線）。

：消除畫面。

S ：將畫完成功的圖存至磁片。

G ：將已存在磁片的圖形表取出到畫面上來。

! ：結束程式。

圖 2—3 造表程式使用法

執行S或G動作時，須要等上一段時間，這是APPLE 正在作圖形表換算的時間。程式 2 — 0 做的工作只是圖 2 — 2 ①～⑦的部份而已，也就是說，每次存至磁片的，只是單一圖形資料而已，接下來，我們仍需將各個圖形連接起來，並且加上開頭的索引，這些工作，我們以程式 2 — 1 來完成：

```
]LIST

10  REM **********************
20  REM **  TABLE  MAKER  **
30  REM **********************
40  REM
50  DIM NA$(60),LN(60)
60  REM
100  REM  ----------
101  REM     MAIN
```

```
102  REM  ----------
110  GOSUB 300: GOSUB 400
120  GOTO 200
200  REM  -- GET SHAPE NAMES --
210  HTAB 1: VTAB 1: PRINT S$(1);
215  HTAB 1: VTAB 2: PRINT S$(2);
220  HTAB 1: VTAB 3: PRINT S$(1);
230  HTAB 13: VTAB 4: PRINT S$(3);
250 X = 1:Y = 6:N = 1
255  IF N = 61 THEN  GOSUB 500: GOTO
     700
258  HTAB X: VTAB Y: PRINT N;".."
260  HTAB X + 3: VTAB Y: INPUT NA$(N
     )
261  IF NA$(N) = "" THEN  GOTO 260
262  IF NA$(1) = "END" THEN  HOME : END

264  IF NA$(N) = "END" THEN N = N -
     1: GOSUB 500: GOTO 700
266 NH = N
268 N1 = N1 + 1:N = N + 1
269 Y = Y + 1
270  IF N1 = 15 THEN N1 = 0:X = X +
     10:Y = 6
275  GOTO 255
300  REM  --- INITIALIZE ---
305  TEXT : HOME : SCALE= 1
310 D$ =  CHR$ (13) +  CHR$ (4)
320  RETURN
400  REM  --- DATA ---
410  DATA   "=========================
     ================"
420  DATA   "              TABLE MAKER

425  DATA   "ENTER SHAPES"
428  REM
```

```
430  FOR I = 1 TO 3: READ S$(I): NEXT

450  RETURN
500  REM  -- READ TO ARRAYS --
505 N = NH
510  HTAB 1: VTAB 21: PRINT "TABLE N
     AME..";
512  HTAB 13: VTAB 21: INPUT NA$
513  IF NA$ = "" THEN 510
514  IF  LEN (NA$) > 6 THEN  GOTO 51
     0
516  HTAB 1: VTAB 22: PRINT "START A
     DDR..";
518  HTAB 13: VTAB 22: INPUT SA$
520  IF SA$ = "" GOTO 518
522 SA =  VAL (SA$)
524  IF SA < 17000 OR SA > 38000 THEN
      GOTO 518
530 SS = (N + 1) * 2:A1 = SA + SS
535  ONERR  GOTO 585
540  FOR I = 1 TO N
542  HTAB 1: VTAB 1
545  PRINT D$;"BLOAD ";NA$(I);",A";A
     1
550 LN(I) =  PEEK ( - 21920) +  PEEK
     ( - 21919) * 256
560 A1 = LN(I) + A1
570  NEXT
575  GOSUB 600
580  POKE 216,0
584  RETURN
585  REM  -- ERROR --
586  HTAB 1: VTAB 22: PRINT "FILE";N
     A$(I);" NOT FOUND--REENTER   ";

587  HTAB 30: VTAB 22: INPUT NA$(I)
```

```
588 NH = N:N = 61: POKE 216,0: GOTO
    255
600  REM  --- CREATE INDEX ---
602 LN(0) = 0:HA = SA:LN = A1 - HA
605  POKE SA,N: POKE SA + 1,0
608 SA = SA + 2
610  FOR I = 0 TO N - 1
620 SS = SS + LN(I)
630 SH =  INT (SS / 256)
635 SL = SS - (SH * 256)
640  POKE SA,SL: POKE SA + 1,SH
650 SA = SA + 2
660  NEXT
670  PRINT D$;"BSAVE ";NA$;', A";HA;
    ",L";LN
699  RETURN
700  REM  -- SHAPE SHOW --
705  HCOLOR= 3:X = 0
710  HTAB 1: VTAB 21: PRINT "WOULD Y
    OU LIKE TO SEE THE SHAPES (Y/N)
    ";: GET GT$
720  IF GT$ = "Y" THEN  HTAB 1: VTAB
    22: PRINT "            PRESS
    SPACE BAR
730  IF GT$ <  > "Y" THEN  TEXT : GOTO
    200
740 I2 =  INT (HA / 256):I1 = HA - (
    I2 * 256)
745  POKE 232,I1: POKE 233,I2
750  GET GT$
755  IF  ASC (GT$) = 27 THEN  TEXT :
     GOTO 200
760  IF X = N THEN X = 0
770  IF  ASC (GT$) <  > 32 THEN  GOTO
    750
775 X = X + 1
```

```
780  HGR : DRAW X AT 140,80: GOTO 75
     0
800  REM  ===============
```

程式2—1　圖形表檔案編輯程式

程式中，522行～570行便是做排列圖形及加上索引等工作的部份。程式2—1執行時字幕會以圖形號碼及“ENTER SHAPES”來要求你鍵入圖形檔案名稱，你可以將由程式2—0所造出的圖形檔案名稱依照您的需要一一加以排列回答，若全部圖形檔名已排列完畢，則以END當成檔案名稱鍵入。程式一接到END訊號，便自動的將各個圖形檔案載入並依次排列起來，最後加上開頭索引及詢問你圖形表名稱，這時你回答的名稱便是眞正的圖形表的檔案名稱了。這時，你還可以將已建好表的各個圖形，依照排列次序一一的顯示在畫面上呢！

2-2 磁帶使用者的自動造表程式

由於磁碟機的檔案存取方便，因此可以先造出各個圖形檔案，然後整編成完整的圖形表。但是對於磁帶的使用者來說，却是無法如此方便使用了，我們的對策是，以畫面第2頁區的8K位址來存放圖形資料（等於是存放於磁碟的效果），然後在圖形表整編的時候，再移往指定的圖形表位址區，也就是說以8K的位址來模擬磁片動作。在程式方面，我們是希望將程式2—0及2—1合併起來，而程式2—1的功能是將圖形總數算出，存入2

7000位址內，然後由27002 開始存入距離索引，最後再接上各組圖形資料，我們只要將這些功能加入到程式 2 — 0 ，便可以省去程式 2 — 1 了。至於模擬磁片的工作則更爲簡單，只要以一組矩陣變數來記錄各圖形資料的起始位址及長度便可以了。圖形資料的移位搬動，是由存於 1800行 的副程式來做的。

```
]LIST

10  REM  ************************
20  REM  **    SHAPE   MAKER    **
30  REM  ************************
35  LOMEM: 30000
40 AV = 16384:K = 0
60  DIM VV$(8),VS%(505),SR$(8)
70  DIM DQ(8),RS%(10),BS$(8),AR(255)
75 AR(0) = AV
80  REM  ====POKE COUSOR====
81  ROT= 0: SCALE= 1
82  FOR I = 0 TO 5: READ WW
84  POKE 768 + I,WW: NEXT I
85  DATA   1,0,4,0,4,0
90  POKE 232,0: POKE 233,3
91  REM ====POKE WRITE ROUTINE====
92  FOR I = 0 TO 44: READ WW
93  POKE 800 + I,WW: NEXT I
94  DATA 169,30,133,60,169,3,133,61,1
     69,31,133,62,169,3,133,63,32,205
     ,254
95  DATA  173,252,3,133,60,173,253,3,
     133,61,173,254,3,133,62,173,255,
     3,133,63,169,32,32,207,254,96
100  REM  ----------------
101  REM         MAIN
```

```
102  REM  ----------------
105  GOSUB 2100: GOSUB 2000
110  GOSUB 2050: GOTO 200
120  REM
200  REM  -- GET KEY --
205  HTAB 20: VTAB 24:XX =  PEEK ( -
     16384)
206  IF XX < 128 THEN  XDRAW 1 AT X,Y
     : XDRAW 1 AT X,Y: GOTO 205
207 A$ =  CHR$ (XX - 128): POKE  - 16
     368,0
208  PRINT A$;
210  IF A$ = "R" THEN  GOSUB 400
215  IF A$ = "E" THEN  GOSUB 500
220  IF A$ = "D" THEN  GOSUB 600:K =
     K + 1:AR(K) = AV: HGR : GOTO 205

225  IF A$ = "S" THEN  GOSUB 1800
230  IF A$ = "#" THEN N = 0: HGR : VTAB
     21: HTAB 34: PRINT "    ": GOTO
     205
240  IF A$ = "!" THEN  TEXT : HOME : END

249  GOSUB 300: GOTO 205
250  REM  -- UP VECTORS? --
255  IF A = 2 THEN SW = SW + 1
260  IF SW = 3 THEN SW = SW - 2: GOTO
     285
270  IF SW = 2 AND A > 4 THEN SW = 0:
      GOTO 285
275  IF A = 2 THEN  RETURN
280 SW = 0: RETURN
285 VS%(N) = 1:N = N + 1:VS%(N) = 3:N
      = N + 1:VS%(N) = A
299  RETURN
300  REM  --- VECTOR PLOT ---
```

```
305 A =  VAL (A$)
308  IF A < 1 OR A > 8 THEN  RETURN
310  IF N + 1 > VL THEN MS = 4: GOSUB
     1500: RETURN
312 N = N + 1:VS%(N) = A
314  GOSUB 250
320  HTAB 35: VTAB 21: PRINT N;
325  IF A > 4 THEN  HPLOT X,Y
330 X = X + X(A):Y = Y + Y(A)
335  RETURN
350  REM  --- VECTOR ERASE ---
355 A = VS%(N)
360 X = X + ( - X(A)):Y = Y + ( - Y(A
     ))
365  HPLOT X,Y
385 N = N - 1: HTAB 35: VTAB 21: PRINT
     N;
399  RETURN
400  REM  -- REPEAT --
404 AH$ = A$
406  HTAB 1: VTAB 22
410  INVERSE : PRINT SR$(3);: NORMAL

412 HB = 8:CH = 3: HTAB 7: VTAB 22: INPUT
     IN$
414 RV =  VAL (IN$)
416  INVERSE :HB = 20:CH = 10: HTAB 1
     0: VTAB 22: INPUT "VECTOR?";IN$:
      NORMAL
420  REM  -GET REPEAT VECTORS-
422 Z =  LEN (IN$)
425  FOR I = 1 TO Z
430 RS%(I) =  VAL ( MID$ (IN$,I,1))
435  NEXT
449  REM  -- VALID VECTOR --
450  FOR I = 1 TO Z
```

```
452  IF RS%(I) > 0 AND RS%(I) < 9 THEN
      NEXT I: GOTO 460
454 MS = 2: GOSUB 1500:I = Z
456  NEXT I: GOTO 499
459  REM  -- PLOT REPEAT --
460  FOR H = 1 TO RV
462  FOR J = 1 TO Z
464 A$ =  STR$ (RS%(J)): GOSUB 300
466  NEXT J: NEXT H:Z = 0
499 A$ = AH$: RETURN
500  REM  -- ERASE --
505  HTAB 1: VTAB 22
510  INVERSE : PRINT SR$(2);: NORMAL

515 HB = 8:CH = 3: HTAB 7: INPUT IN$
520  IF NF = 1 THEN  RETURN
525 EE =  VAL (IN$)
530  HCOLOR= 0
560  IF EE = 0 THEN EE = 1
565  FOR I = 1 TO EE
570  IF N = 0 THEN I = EE: GOTO 590
585  GOSUB 350
590  NEXT I
595  HTAB 35: VTAB 21: PRINT "   ";
596  HTAB 35: VTAB 21: PRINT N;
599  HCOLOR= 3: RETURN
600  REM  --- SAVE SHAPE ---
602  IF N = 0 THEN  RETURN
604 C =  FRE (0)
628 VA = AV
630  HTAB 15: VTAB 22: FLASH : PRINT
     "PLEASE WAIT";: NORMAL
635 AA = 0:BB = 0:CC = 0
640  FOR I = 1 TO N
650 J = VS%(I): GOSUB 700
660  NEXT
```

```
682  IF AA = 0 THEN  GOTO 689
684  IF BB = 0 THEN B$ = "000"
686 C$ = "00"
688  GOSUB 800
689  POKE AV,0:AV = AV + 1
690  PRINT AV
691 N = 0
692  VTAB 21: HTAB 34: PRINT "     "
695  RETURN
700  REM  -COMBINE VECTORS-
705  IF AA = 1 THEN  GOTO 720
710 A$ = VV$(J):AA = 1: GOTO 799
715 PA = J
720  IF BB = 1 THEN  GOTO 740
725 PB = J
730 B$ = VV$(J):BB = 1: GOTO 799
740 C$ =  MID$ (VV$(J),2,2)
750  IF J = 2 OR J > 4 THEN C$ = "00"
     :I = I - 1
760  IF PB = 2 AND (J = 2 OR J > 4) THEN
     B$ = "000":I = I - 1
780  GOSUB 800
799  RETURN
800  REM  ----- BINARY CONV -----
810 D$ = C$ + B$ + A$
820  FOR IQ = 1 TO 8
830 DQ(IQ) =  VAL ( MID$ (D$,IQ,1))
835  NEXT IQ
840 AD = DQ(1) * 128 + DQ(2) * 64 + D
     Q(3) * 32 + DQ(4) * 16 + DQ(5) *
     8 + DQ(6) * 4 + DQ(7) * 2 + DQ(8
     )
842  POKE AV,AD:AV = AV + 1
845 AA = 0:BB = 0:CC = 0
848  RETURN
850  RETURN
```

```
1500  REM  -- ERROR --
1510  HTAB 1: VTAB 23
1520  PRINT SR$(4 + MS);
1530  GET E$
1540  IF  ASC (E$) = 13 THEN 1530
1545 H = RV:J = Z
1550  HTAB 1: VTAB 23: PRINT SR$(5);
1599  RETURN
1800  REM ====BUILD SHAPE TABLE AND S
      AVE====
1801  HOME : VTAB 22: INPUT "SAVE TO
      WHERE(ADDRESS)?";AR
1802  POKE AR,K: POKE AR + 1,0:SA = A
      R + 4
1804  HOME : VTAB 22
1805 SS = 2 + 2 * K: POKE AR + 2,SS -
       INT (SS / 256) * 256: POKE AR +
      3, INT (SS / 256)
1806  FOR I = 0 TO K - 1:SS = SS + AR
      (I + 1) - AR(I)
1807 SH =  INT (SS / 256)
1808 SL = SS - (SH * 256)
1809  POKE SA,SL: POKE SA + 1,SH
1810 SA = SA + 2: NEXT I
1819 A1 = AR + 2 + 2 * K
1820  FOR KK = 0 TO K - 1
1830  FOR J = AR(KK) TO AR(KK + 1) -
      1
1840  POKE A1, PEEK (J)
1850 A1 = A1 + 1
1860  NEXT J: NEXT KK
1870 LL = AR(K) - 16384 + 2 + 2 * K: POKE
      798,LL -  INT (LL / 256) * 256: POKE
      799, INT (LL / 256)
1875 DE = AR + LL
1880  POKE 1020,AR -  INT (AR / 256) *
```

```
     256: POKE 1021, INT (AR / 256): POKE
     1022,DE -  INT (DE / 256) * 256:
      POKE 1023, INT (DE / 256)
1890  TEXT : HOME : PRINT "PLEASE PRE
     SS 'PLAY & RECORD'"
1891  PRINT "OF THE TAPE RECORD. ": INVERSE
     : PRINT "HIT ANY KEY TO BEGIN": GET
     B$: CALL 800
1900  RETURN
2000  REM  -- INIT --
2004 AR = 17000:VL = 500
2010  HCOLOR= 3:NM = 1:UC = 0:LM = 50
     0
2040 X = 140:Y = 80:DC = 128
2045  HOME : HGR : SCALE= 1
2049  RETURN
2050  REM  -- SETUP
2055  HTAB 1: VTAB 21: PRINT SR$(4);
2060  HTAB 1: VTAB 23: PRINT SR$(5);
2070  RETURN
2100  REM  --- DATA ---
2102  DATA  -1,0,0,-1,1,0,0,1
2104  DATA  -1,0,0,-1,1,0,0,1
2110  DATA  "NAME
                "
2120  DATA  "NUMBER
              "
2124  DATA  "NUMBER       VECTOR
                       "
2126  DATA "SAVE      REPEAT          #C
     LEAR"
2128  DATA  "DONE      ERASE
     END      1-8/PLOT"
2132  DATA   "INVAID VECTOR
                      "
2134  DATA  "FILE NOT FOUND
```

```
                          "
2136  DATA  "VECTOR LIMIT REACHED
                          "
2140  DATA  "011","000","001","010"
2142  DATA  "111","100","101","110"
2148  REM  ---- READ DATA ----
2150  FOR I = 1 TO 8: READ X(I),Y(I):
      NEXT
2170  FOR I = 1 TO 8: READ SR$(I): NEXT

2180  FOR I = 1 TO 8: READ VV$(I): NEXT

2199  RETURN

]
```

程式 2－2　磁帶的自動造表程式

```
*320L

0320-   A9 1E       LDA   #$1E
0322-   85 3C       STA   $3C
0324-   A9 03       LDA   #$03
0326-   85 3D       STA   $3D
0328-   A9 1F       LDA   #$1F
032A-   85 3E       STA   $3E
032C-   A9 03       LDA   #$03
032E-   85 3F       STA   $3F
0330-   20 CD FE    JSR   $FECD
0333-   AD FC 03    LDA   $03FC
0336-   85 3C       STA   $3C
0338-   AD FD 03    LDA   $03FD
```

```
033B-   85 3D       STA   $3D
033D-   AD FE 03    LDA   $03FE
0340-   85 3E       STA   $3E
0342-   AD FF 03    LDA   $03FF
0345-   85 3F       STA   $3F
0347-   A9 20       LDA   #$20
0349-   20 CF FE    JSR   $FECF
034C-   60          RTS
```

程式 2 － 3　寫圖形表到磁帶的副程式

程式 2 — 2 執行時，其操作法與程式 2 — 0 相似，但是沒有“ G ”命令，並且每次畫完一個圖形之後可以按下“ D ”鍵，這時圖形便會被編碼並存入到 $4000 以後的位址內。如果另外還有圖形，你可以按下“ # ”將目前圖形消除，再開始畫新圖形。全部圖形畫好之後，按下“ S ”，APPLE 便自動將存在 $4000 的圖形資料加上索引，然後問你圖形表要存放的位址，回答了存放位址後，APPLE 便 CALL 程式 2 — 3 的副程式來將圖形表寫入磁帶。

用程式 2 — 2 造出的圖形表使用法有二：一是馬上寫入主程式（記得NEW），這時 ($E8 , $E9)是必須指向你回答的位址，才能畫圖形。另一個方法是當以後要使用圖形表時，以 SHLOAD 指令來讀取已經由程式 2 — 2 造好存入至磁帶的圖形表。這種方式，（ $E8 , $E9 ）已自動被指向圖形表位址（是 APPLE 最高可用RAM區域，而不是你回答程式 2 — 2 的位址），你只要直接使用 ROT，SCALE，XDRAW 便可以了。

第 3 章

戲劇效果的文數字形圖形表

		ALPHA	SHAPE	TABLE
1.	13.,	25.8	37.D	49.P
2.!	14.-	26.9	38.E	50.Q
3."	15..	27.:	39.F	51.R
4.#	16./	28.;	40.G	52.S
5.$	17.0	29.←	41.H	53.T
6.%	18.1	30.=	42.I	54.U
7.&	19.2	31.→	43.J	55.V
8.'	20.3	32.?	44.K	56.W
9.(	21.4	33.■	45.L	57.X
10.)	22.5	34.A	46.M	58.Y
11.*	23.6	35.B	47.N	59.Z
12.+	24.7	36.C	48.O	

3-0 字形圖形表

雖然是GAME的程式，但也常有需要使用文字、數字的場合，可是APPLE 本身的字形在圖形文字混合的時候，只能印出在畫面底下四行，並且字體呆板，沒有變化。因此，我們便來造出較美觀、較戲劇化、並且可以在高解像畫面任意位置上顯示的高解像字形吧！

```
]CALL-151

*61A8.6695

61A8- 3B 00 78 00 7C 00 87 00
61B0- 97 00 B1 00 C5 00 D5 00
61B8- EB 00 F8 00 05 01 10 01
61C0- 24 01 32 01 3B 01 47 01
61C8- 50 01 5A 01 73 01 81 01
61D0- 9C 01 B3 01 CD 01 E7 01
61D8- 03 02 14 02 2E 02 4A 02
61E0- 57 02 65 02 74 02 86 02
61E8- 96 02 A8 02 D2 02 E7 02
61F0- FE 02 11 03 26 03 3D 03
61F8- 53 03 6A 03 80 03 8E 03
6200- A3 03 B9 03 CC 03 E3 03
6208- F9 03 0D 04 20 04 37 04
6210- 4E 04 5F 04 71 04 89 04
6218- 9C 04 B5 04 C7 04 DA 04
6220- 49 49 01 00 25 27 28 3C
6228- 2C 3C 2C 3C 2C 04 00 01
6230- C0 40 18 24 24 35 36 6E
6238- 21 24 2C 36 36 06 00 01
6240- C0 08 6D 2D 38 3F 3F 27
6248- 2D 2D 2D 1C E1 3F 28 2D
6250- 2D 25 3F 3F 3F 0C 6D 2D
6258- 00 49 05 38 3F 07 28 0D
6260- 2D 1C 27 3C 3E E0 6D 2D
6268- 1C 3F 67 31 00 49 29 3C
6270- DF 2B 2C 2C 2C 2C 2C 2C
6278- DC 1B 37 35 00 29 6D E5
6280- FF 3B 2C 0D 2D 05 F8 FF
6288- 27 6D E5 3F 3C 2C 2C 35
6290- 35 3F 00 01 C0 40 18 05
6298- 28 38 07 28 25 3F 07 00
62A0- 49 39 3C 3C 3C 2C 3C 2C
62A8- 2C 2C 2C 04 00 25 25 25
62B0- 25 27 25 27 27 27 27 00
62B8- 09 40 24 24 24 8D 1E 1E
62C0- 1E 1E 04 28 2D B5 07 38
62C8- 38 38 38 00 41 25 27 25
62D0- 2D 3C BF 3F 2C 25 25 27
62D8- 25 00 05 28 38 07 28 25
```

```
62E0- 3F 04 00 01 C0 01 C0 2D
62E8- 2D 2D 3C 3F 3F 3F 00 29
62F0- 05 38 3F 2C 2D 1C 3F 00
62F8- 01 C0 25 25 25 25 25 25
6300- 25 00 29 2D 05 20 24 24
6308- 24 1C 3F BF 36 36 36 2E
6310- 2D 25 24 24 24 3F 37 36
6318- 36 36 00 2D 25 3F 67 25
6320- 27 25 27 25 27 27 2D 36
6328- 00 2D 2D 25 3F 3F 27 6D
6330- 01 38 3F 67 2D 2D 24 24
6338- 1C 3F BF 2D 2D 36 FE 03
6340- 38 2C 05 00 29 2D 05 38
6348- 3F 3F 2C 4D 25 27 3F 2C
6350- 2D 38 2C 3C 3F 37 27 0C
6358- 2D 2D 00 49 29 3C 2C 05
6360- 38 3F 3F 27 2D 2D 2D 1C
6368- 27 25 27 25 3F 17 17 17
6370- 2E 2C 28 28 00 29 2D 05
6378- 38 3F 3F 2C 4D 25 27 25
6380- 3F 3F 07 28 2D E5 1B 27
6388- 2D 2D 25 3F 3F 3F 00 29
6390- 2D 05 38 3F 3F 2C 3C 2C
6398- 3C 2C 3C 0C 67 2D AD 3F
63A0- 3F 4E 29 D6 3F 2E 2D 3E
63A8- 2E 05 00 29 3C 2C 2C 2C
63B0- 2C 3C 2C 3C 0C 3F 3F 37
63B8- 2E 2C 2D 00 29 2D 05 38
63C0- 3F 3F 2C 3C 0C 2D 25 3F
63C8- 3F 28 3C 2C 2C 2E 2C 2E
63D0- 3E 2E 96 37 2D 00 29 2D
63D8- 05 38 3F 3F 2C 4D 25 67
63E0- 3F 3F 07 28 2D 2D 3C 2C
63E8- E4 3F BF 2D 2D DE 3B 2E
63F0- 05 00 C1 2D 3C 27 2D C4
63F8- 08 3F 2C 25 3F 07 00 05
6400- 28 38 07 28 25 3F 04 28
6408- 25 3F 2C 2D 00 C1 08 18
6410- 20 0C 36 76 24 24 AC 32
6418- 25 35 05 00 01 C0 2D 2D
6420- 2D 3C 3F 3F E7 40 2D 2D
6428- 2D 3C 3F 3F 3F 00 C1 08
6430- 18 20 35 25 25 AC 36 2E
6438- 20 96 1B 24 04 00 09 25
6440- 27 28 3C 0C 25 25 27 25
```

```
6448- 1C 3E 3C 3E BC 2E 05 00
6450- 23 24 24 24 24 2D 2D 2D
6458- 35 36 36 36 36 3F 3F 3F
6460- 24 24 24 24 2D 2D 35 36
6468- 36 36 3F 3F 24 24 24 2D
6470- 35 36 36 3F 24 24 35 36
6478- 06 00 25 27 25 27 25 67
6480- 25 0C 2E 2E 15 37 35 3F
6488- 3F 4E 29 3E 2E 06 00 2D
6490- 2D 2D 38 2C 3C 3C 67 29
6498- 3C 2C 1C 3F 3F 77 35 37
64A0- 35 37 35 37 35 00 09 2D
64A8- 2C 2C DC 9B 3A 3C 2C 3C
64B0- 2C 3C 0C 25 2D 2E 2E 05
64B8- 00 2D 3C 2C 3C 2C 3C 2C
64C0- 3C 07 28 2D 35 35 35 37
64C8- 35 37 F5 37 3F 00 2D 3C
64D0- 2C 3C 2C 3C 2C 3C 1C 2D
64D8- 2D 2D 17 35 DE 17 35 4E
64E0- 17 35 3F 3F 00 2D 3C 2C
64E8- 3C 2C 3C 2C 3C 1C 2D 2D
64F0- 2D 17 35 DE 17 35 4E 9A
64F8- 1B 07 00 09 2D 05 38 0C
6500- 25 3F B7 3B 3C 2C 3C 2C
6508- 3C 0C 25 2D 2E 2E 96 12
6510- 3E 00 24 24 24 24 35 36
6518- 2E 2D 25 24 2C 36 36 3E
6520- 2E 3E 0E FF 1B 24 24 00
6528- 09 2D E5 27 25 27 25 27
6530- 25 3F 28 2D 05 00 01 20
6538- 2C 36 2D 2D 24 24 24 07
6540- 28 2D 1E 36 36 36 1E 3F
6548- 3F 07 00 2D 3C 2C 3C 2C
6550- 3C 2C 3C 07 28 6D 29 3E
6558- 2E 1E BF 35 35 35 37 35
6560- 00 2D 3C 2C 3C 2C 3C 2C
6568- 3C 07 28 2D 96 92 12 2D
6570- 25 BC 07 00 25 27 25 27
6578- 25 27 25 27 AD B6 21 0C
6580- 64 35 37 35 37 35 37 35
6588- 37 35 00 25 27 25 27 25
6590- 27 25 27 AD 2E 2E 2E 36
6598- 36 25 24 24 24 3C 36 36
65A0- 00 09 3C 3C 2C 3C 2C 3C
65A8- 0C 25 2D 2E 2E 3E 2E 3E
```

```
65B0- 2E 1E BF 2D 00 2D E5 27
65B8- 25 27 25 27 25 3F 28 2D
65C0- 2D 2E 3E 2E 1E 3F 0E 00
65C8- 49 24 3F 3C 2C 3C 2C 3C
65D0- 0C 2D 2D 2E 3E 2E 3E 2E
65D8- 1E 27 18 76 32 2D 00 2D
65E0- E5 27 25 27 25 27 25 3F
65E8- 28 2D 2D 2E 3E 2E 1E 3F
65F0- 0E 35 35 37 35 00 29 2D
65F8- 05 38 DF 67 49 E5 27 27
6600- 27 27 25 2D 35 2D 00 29
6608- 2D 1C 27 25 27 25 27 25
6610- 3F 1E 24 2D 2D 35 3E 20
6618- 00 29 2D 2D 38 3F 3F 27
6620- 25 27 25 27 25 27 6D 09
6628- 35 37 35 37 35 2F 3E 07
6630- 00 49 3C 3C 3C 2C 3C 2C
6638- 3C 2C 4D 29 3E 2E 3E 2E
6640- 3E 3E 2E 00 09 24 17 07
6648- 28 3C 2C 3C 2C 3C 2C 4D
6650- 29 3E 2E 3E 2E 3E 2E 1E
6658- 1E 24 27 24 00 25 67 25
6660- 25 3C 3C 3C 2C 4D 29 3E
6668- 3E B6 2E 2E 3E 07 00 29
6670- 2D 1C 27 25 27 3F 20 24
6678- 35 36 2D 25 24 35 36 1E
6680- 3F 00 25 27 25 25 25 25
6688- DF 2B 38 2C 2D 2D 1E 35
6690- 96 BA 35 3F 3F 00
```

程式 3 － 0 文數字形的圖形表（請存至磁片或磁帶，檔名：ALPHA-TBL）

3-1 字形圖形表的使用法

程式3 — 0的文數字形表，如本章篇頭的畫面所示，共有59個字形，第一個是空格，最後一個是Z。這些文數字形在本書往後的程式均要使用到，因此請CALL-151，然後61A8：3B 00

78 00 7C ……將字形表鍵入，當然也要以61A8·6695W存錄到磁帶上。當以後的程式必須使用到這組圖形表的時候，只要以CALL-151 進入監督程式，並且以61A8·6695將之讀取，便可加以使用了。

像文數字形這種常用的圖形表，最好給予固定的位址，以方便在BASIC 主程式裡頭使用，此處的$61A8 ，其高位元組$61＝97，低位元組$A8＝168，因此每次我們要叫用這組圖形表時，便要如前章所述的先POKE 232,168:POKE 233,97然後DRAW LT AT X,Y便可以在（X，Y）位置上畫出第LT號的字形出來了。在多數程式裡，我們另有其它圖案的圖形表，這些圖案圖形表，我們將之安排在由$6978 開始的位址，因此畫完了文數字形，要改畫圖案時，必須POKE 232,120:POKE 233,105，才能把字形圖形表改換成圖案圖形表。

字形圖形表的使用例

```
]LIST

0  REM  ************************
1  REM  ** ALPHA SHAPE TEXT  **
2  REM  **      ON PAGE-2     **
3  REM  ************************
9  HGR2
10  POKE 232,168: POKE 233,97
15 I = 0
20  FOR X = 10 TO 270 STEP 20
30  FOR Y = 10 TO 180 STEP 20
```

```
40 I = I + 1
45  IF I > 59 THEN  END
50  DRAW I AT X,Y
60  NEXT Y: NEXT X
```

程式 3—1　字形圖形表使用例（請先存入程式 3—0 才可執行）

```
0  REM  ************************
2  REM  **   LETTER  DEMO   **
4  REM  ************************
7  PRINT  CHR$ (4)"BLOAD ALPHA-TBL"
8  HOME : ROT= 0: SCALE= 1
9  INVERSE
10  INPUT "PLEASE INPUT YOUR NAME !!!";NA$
11  NORMAL
12  HGR
15 LX = 10:LY = 10
19  POKE 232,168: POKE 233,97
20  FOR I = 1 TO  LEN (NA$)
30 SV =  ASC ( MID$ (NA$,I,1))
40  DRAW SV - 31 AT LX,LY
50 LX = LX + 9
60  NEXT : VTAB 22: GOTO 10
```

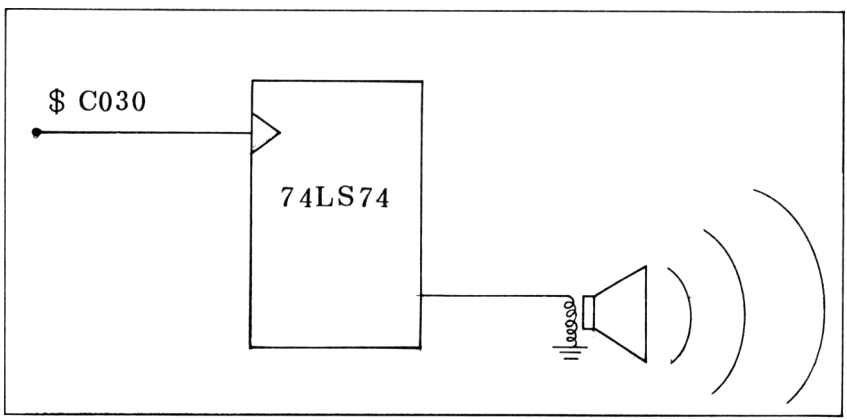
程式 3—2　另一個例子（請先存入程式 3—0 的圖形表）

程式 3 — 1 是將字形圖形表的內容一一印出來，只要指定位置及圖號，便可以印出所需要的文字了。程式 3 — 2 則是一個更實用的例子，它可以將 NA$ 字串的內容以字形圖形表的方式畫

到APPLE 畫面上來，其方式是利用 MID 函數將 NA$ 字串的每一個字分離出來，經由 ASC 函數取其ASCII碼，減去 31 之後，就是該字所對應的圖形的圖號了，譬如“ B ”的ASCII碼是 66 ，減掉 31 之後是 35 ，你如果下DRAW 35 AT 50,50 便可以在高解像畫面(50,50)的位置上畫出一個“ B ”字來。因此程式 3 — 2 的 10000～10080行便是一個畫出字串字形的副程式，你可以設定 NA$ 的值，然後GOSUB 10000便可以畫出各種字形來了。程式 3 — 2 執行時，每次你鍵入一組字串，APPLE 便自動的將之畫出在畫面上，如果配合XDRAW 的消圖功能，更可以造出閃爍，反白的效果呢！請儘量的利用這個程式的優點來使你的畫面更生動更有趣吧！

第 4 章

音響的效果



遊戲程式注意的，除了畫面有趣的色彩及動作之外，便該是聲音的效果了。此起一些其他的微電腦，譬如ATARI—400 或VIC—20、64 等，Apple的硬體發聲結構是相當單純的，在Apple的路線上，它只用一個位址 $C030 來觸動喇叭，如果每秒鐘觸動的次數多，喇叭便振動快而發出頻率較高的聲音，如果觸動的次數少，喇叭便振動慢而發出較低的音調。除了使用另外的界面版或是卡帶輸出口（ $C020)之外，大多數的聲音程式均是以觸動 $C030 這個位址，而造出奇妙的音響來。現在，就來看看這種效果是如何達成的：

4-0 一個發聲的例子

下例便是本書主要的發聲程式，它可以附加到任何程式當中使用，其使用法如例中旁白說明所示，程式執行時，首先 GOSUB 3000 將發聲音的機械語言副程式讀入到 $300 開始的第 3 頁區，然後您便可以在主程式當中呼叫使用了。

```
]LIST

10  GOSUB 3000: REM  GO SOUND ROUTINE
20  REM  ****************************
30  REM  **  MAIN PROGRAM BEGIN  **
40  REM  **         HERE !       **
50  REM  **     ............     **
60  REM  **     ............     **
70  REM  **     ............     **
80  REM  ****************************
100  END
3000  REM   ================
3030  REM      POKE  SOUND
3040  REM   ================
3050  FOR I = 0 TO 22
3060  READ WW
3070  POKE 768 + I,WW
3080  NEXT I: RETURN
3090  DATA 169,10,  133,1,173,48
3100  DATA 192,136,208,4,198,1
3110  DATA 240,8,202,208,246
3120  DATA 166,0,76,4,3,96
```

主程式部份

聲音副程式

程式 4—0　發聲程式

```
*300L

0300-   A9 0A       LDA   #$0A
0302-   85 01       STA   $01
0304-   AD 30 C0    LDA   $C030
0307-   88          DEY
0308-   D0 04       BNE   $030E
030A-   C6 01       DEC   $01
030C-   F0 08       BEQ   $0316
030E-   CA          DEX
030F-   D0 F6       BNE   $0307
0311-   A6 00       LDX   $00
0313-   4C 04 03    JMP   $0304
0316-   60          RTS
```

程式4－1　發聲副程式（檔案名：SOUND）

程式4—1便是發聲副程式的機械語言形態。其中$316是程式結束點的RTS，程式的結束是由$30A～$30D 的DEC $01及BEQ $316 來跳出發聲迴圈，因此改變$01這個位址的內含值便可以令發聲的時間長一點或短一點，$01的內含值是由$300～$303 來設定的，$30E～310 的DEX是用來決定每秒鐘迴圈跑過$304 的次數，也就是觸動$C030 喇叭位址的次數，因此改變X…暫存器的初值就可令喇叭發生較高或較低的音調。所以您可以在主程式裡 POKE 769，音長：POKE 0 ，音高，然後CALL 768 便可令喇叭發生所希望音長及音高的聲音了，請將程式4—0的第50 行改如下例，便可印證我們的說法。

```
50  FOR I = 0 TO 40: POKE 769,I: POKE 0,I: CALL
    768: NEXT
60  REM  **     ............      **
70  REM  **     ............      **
```

程式 4 —2

4-1 另一個聲音程式

發聲的程式很多，但不外乎以控制喇叭被觸動的情況來造出各種音高及音長，現在再舉一例說明。

```
*300L

0300-   AD 30 C0    LDA   $C030
0303-   A5 00       LDA   $00
0305-   20 A8 FC    JSR   $FCA8
0308-   A5 01       LDA   $01
030A-   D0 04       BNE   $0310
030C-   C6 02       DEC   $02
030E-   F0 05       BEQ   $0315
0310-   C6 01       DEC   $01
0312-   4C 00 03    JMP   $0300
0315-   60          RTS
```

程式 4 — 3　另一種程式

程式 4 — 3 的發聲副程式使用了 Apple 內部的一個延時副程式 $FCA8 來控制喇叭觸動的頻率，其延時長度由位址 $00 的內含值所決定，而位址 $02 的內含值則決定所發出聲音持續的長度（即音長），程式 4 — 3 的使用例如下：

```
]LIST

0  REM  ******************************
1  REM  **   SOUND  ROUTINE !!!   **
2  REM  ******************************
5  GOSUB 2000
6 V = 1
7  FOR L = 1 TO 15
10 K = 1: POKE 1,V: POKE 2,K: POKE 0,L
15 V = V + 1
16  IF V > 120 THEN 6
20  CALL 768
30  NEXT L
40  GOTO 7
2000  REM  ===POKEING INTO THE SOUND ROUTINE
     ===
2005  FOR D = 0 TO 21: READ W: POKE 768 + D,
     W: NEXT D: RETURN
2010  DATA  173,48,192,165,0,32
2020  DATA  168,252,165,1,208
2030  DATA  4,198,2,240,5,198
2040  DATA   1,76,0,3,96
```

程式 4 一 4

您可以看出，一個聲音程式的效果，除了其機械語言副程式之外，主程式的安排更是一個重要的部分。

至於主程式是如何安排的呢？下列的例子可以給你一點建議：

```
]LIST

1  REM  ***************************
2  REM  ** APPLE MUSIC PROGRAM **
3  REM  ***************************
5  LOMEM: 30000
10  HOME
20  VTAB 10: HTAB 12: FLASH : PRINT "
     APPLE MUSIC...."
25  INVERSE : VTAB 20: HTAB 14: PRINT
     "HIT ANY KEY"
30  IF  PEEK ( - 16384) < 128 THEN 30

40  NORMAL
50  DIM A$(255),B$(255)
60  FOR I = 0 TO 21: READ WW
70  POKE 770 + I,WW: NEXT I
100  DATA  173,48,192,165,0
110  DATA  32,168,252,165,1
120  DATA  208,4,198, 2,240
130  DATA  5,198,1,76,2,3,96
200 A$ = "00110071521172017201720152
     1521152110071521100710012"
250  FOR ZZ = 1 TO  LEN (A$)
260 B$(ZZ) =  MID$ (A$,ZZ,1)
270  NEXT ZZ
300  FOR Z = 1 TO  LEN (A$) - 3 STEP
     4
310 Z1 = ( ASC (B$(Z)) - 48) * ( ASC
     (B$(Z + 1)) - 48) +  ASC (B$(Z +
     2)) - 48
320 Z2 = 40 - Z1
330  POKE 0,Z2
340  POKE 2, ASC (B$(Z + 3)) - 48
350  POKE 1,75
```

```
360  CALL 770
370  NEXT Z
380  IF  PEEK ( - 16384) > 128 THEN 4
     00
390  IF  PEEK ( - 16384) <  > 141 THEN
     380
```

程式 4—5　音樂程式

```
*302L

0302-   AD 30 C0     LDA   $C030
0305-   A5 00        LDA   $00
0307-   20 A8 FC     JSR   $FCA8
030A-   A5 01        LDA   $01
030C-   D0 04        BNE   $0312
030E-   C6 02        DEC   $02
0310-   F0 05        BEQ   $0317
0312-   C6 01        DEC   $01
0314-   4C 02 03     JMP   $0302
0317-   60           RTS
```

程式 4—5　的發聲副程式

程式中 60 ～ 130 行是用來將聲聲的機械語言副程式讀入至 $302～$317 位址內，這個副程式與前面介紹例子大同小異，此處我們要強調的是，如何適當的來安排令其發出我們所需要的聲音。程式 4 — 5 是利用 330 行與 300 行以改變 $00 及 $02 這兩個位址的內含值來改變其發音的音高與音長，但是，像五線譜這樣變化多端的構造，要如何安排來存入 $00 與$02

呢？我們是將音長與音高以一個字串 A$ 來表示，然後用 B$(Z)=MID$(A$,Z,1) 來取出安排在 A$ 的音高與音長，所以只要改換 A$ 的內容，便可以造出不同的音樂來。那麼音長音高是如何安排的呢？ A$ 以每 4 個數字爲一組，前 3 個數字組成 Z2 變數 POKE 到 $00 ，用來控制音高，而第 4 個數字則 POKE 到 $02，用來控制音長，請改變 A$ 的內容，來造出美妙的音樂吧！

第 5 章

磁碟機的使用者

如何製造您的程式(一)

在這一本書，我們所要介紹的程式主要是以 Applesoft 的圖形表來造出畫面的遊戲效果，因此圖形表可能佔用了很大的一部份，而圖形表當中又以英文字形表爲大部份，這些英文字形的圖形表在每一則程式均是相同的，所以事實上，我們是不需要將這些繁長的資料每次均鍵入到程式裡頭，如果有辦法只造出一次圖形表，然後每則程式均可叫用同樣的這個圖形表，說來當然是方便多了。

所以我們便以第二章的〝BUILD SHAPE TABLE〞的程式來造出各種圖形表，然後以二進位檔的方式BSAVE 到磁片內，每次主程式執行時，再以BLOAD 取回便可以了，其整個記憶體位址的架構如下：

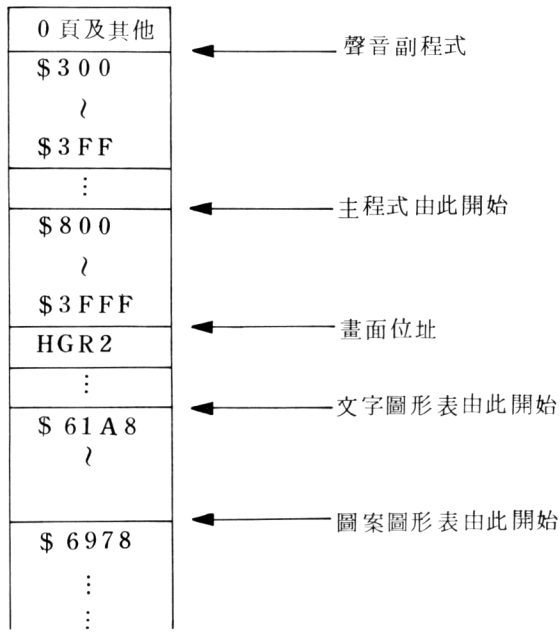
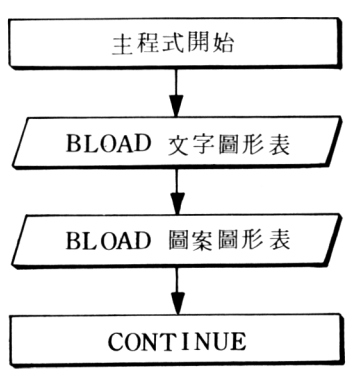


圖 5—0

程式執行的情形如下：



如果您自己設計了圖形表，那麼可以CALL—151 在監督程式的狀態下（＊號）鍵入圖形表資料（請從 61A8 或 6978 開始，以便與本書的程式符合），如果您使用了第二章所介紹的自動造表程式，則資料已自動存好了，說起來也是很方便的。

第 6 章

卡式錄音帶的使用者

如何製造您的程式(二)

對於卡式錄音帶的使用者，由於沒有BLOAD 及BSAVE 等指令，因此不可能由多個程式取用磁片當中共同的一個檔案，但是只要我們能夠將這些共同的資料附加到每則程式上，效果也是一樣的，其實，您會發現，在使用上說不定是更加方便而有趣呢！

不過，問題看似不甚簡單，因爲我們的主程式是以BASIC寫成的，而圖形表資料基本上是機械語言，如果說要使用POKE的方式來存入圖形表資料，則面對着像文字字形這種四、五十種圖形的情形，每次要在將些資料鍵入主程式不是很繁複而且也失去我們原本的目的了嗎？

當我參考了施威銘先生所着APPLE Ⅱ徹底研究第一册第四章的聯結機械語言與BASIC 程式的各種方法之後，最初也似乎沒有任何合理的方法來將BASIC 的主程式與圖形表資料聯結在一起。因爲畢竟本書用到的圖形表是太長了，唯一的方法只有存放在程式之後的高位址區，而如果直接接到程式之後，則由於每個程式長度不一，如此勢必改變每則主程式裡頭的圖形表起始位址指位器的內含值了，並且每次程式一經修改（加長）之後，可能又要移動圖形表，否則當圖形表一旦附加上來，可能就會與BASIC 程式相重疊了。

所以，理想的方式應該是將圖形表資料放在高位址遠離程式與畫面的地方，因此我們的安排仍然是與圖5-0的情形相同，無論您使用的是磁碟機或錄音帶，程式結構都是一樣的。可是，怎樣將BASIC 與圖形表資料一齊存到磁帶上，而又如何讀回APPLE 呢？我們有辦法將BASIC 程式連同在$6978 的圖形表資料一併存入磁帶，但是這樣就必須將夾在BASIC 程式與$61A8之間的無用資料也一併存入磁帶，造成既費時又浪費磁帶的情形，目前有些Game 其實只有4～5K的長度，却需幾分鐘的讀入時間，多半便是這樣造成的。

程式保護法的另一種用途

在APPLE Ⅱ徹底研究第一册裡，有一個例4－2：使用這種方法，可以使程式執行時，在BASIC 狀態下執行監督程式的指令。由此，不禁令人想起：如果將這個程式加以運用，是否也可以如磁碟機的BLOAD、BSAVE 一般的，在程式執行當中以〝61A8、6FFFR〞的方式將圖形表讀取呢？

原則上，這個方法是可行的，問題是，使用BLOAD 讀取資料時，磁碟機會自行轉動而達到資料讀取的目的，而〝61A8．6FFFR〞却不能令卡式錄音機自行轉動以讀取資料，這樣一來勢必要第一次LOAD進來BASIC 程式，然後以上述的方法執行〝61A8．6FFFR〞，並且再按錄音機的PLAY 鍵，以讀進圖形表資料……到最後，你發現這比直接鍵入CALL－151 61A8．6FFFR更麻煩！

在APPLE Ⅱ徹底研究第二册附錄當中，有一個程式保護的

方法，你只要先POKE 82，213　然後將BASIC 程式SAVE到磁帶，如此當下次將程式LOAD回來時，程式便會自動執行，而RESET 後，鍵入任何指令均會被當成RUN　而執行程式。利用這個方法，將BASIC 程式存入到磁帶，接着進入監督程式，將圖形表資料也存入磁帶，這樣便完成我們的手續了！

一旦，你自磁帶以LOAD指令讀入BASIC 主程式，讀取完畢之後，程式便開始自動執行，而程式執行時便先做〝61A8·6FFFR 〞的動作，這時因爲錄音機仍在轉動，所以錄在主程式後面的圖形表資料便被讀取，而讀取完畢之後，接着便開始我們的Game了。

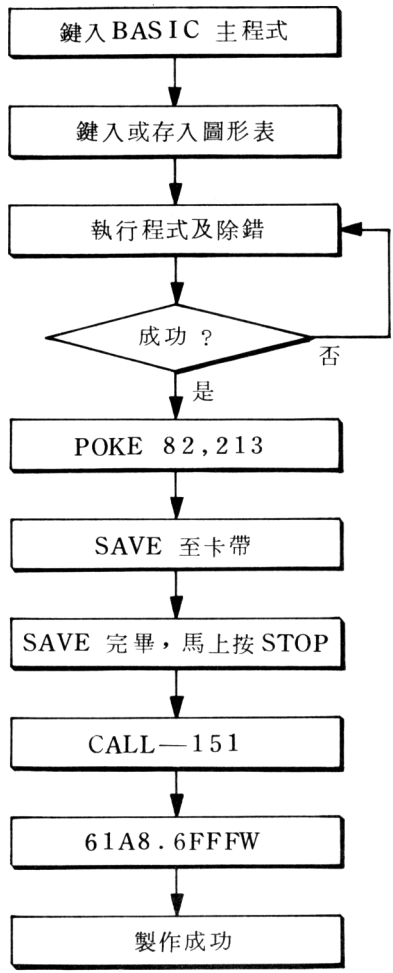


圖 6－0　磁帶的製作方法

要以圖 6－0 的方式成功的來造出磁帶，BASIC 主程式必須是這樣安排才行：

```
]LIST

10  REM  *** AUTO READ ROUTINE ***
20  GOSUB 2000          : REM  將指令讀入第2頁
30  CALL  - 144         : REM  執行第二頁的命令
40  REM  ......................
50  REM  ......................
60  REM  ....MAIN PROGRAM.....
70  REM  ......................
1999  END
2000  REM   -----AUTO READ BEGIN !!!-----
2010  DIM Y$(100)       : REM  將 "61A8.6FFFR"
2011  INVERSE           : REM  取入第2頁
2015  FOR D = 0 TO 500: NEXT D: PRINT "WAIT
     !!! PLEASE !!!": NORMAL
2020 Y$ = "61A8.6FFFR N D823G"
2030  FOR I = 1 TO  LEN (Y$)
2040  POKE 511 + I, ASC ( MID$ (Y$,I,1)) + 1
     28
2050  NEXT I
2060  POKE 72,0
2070  RETURN
```

原程式

程式6－0　自動讀圖形表的程式架構

也就是說在您設計出來的程式之前要加入程式6—0的第20及30行，在程式之後也要接上程式6—0的第1999 行至2070行。

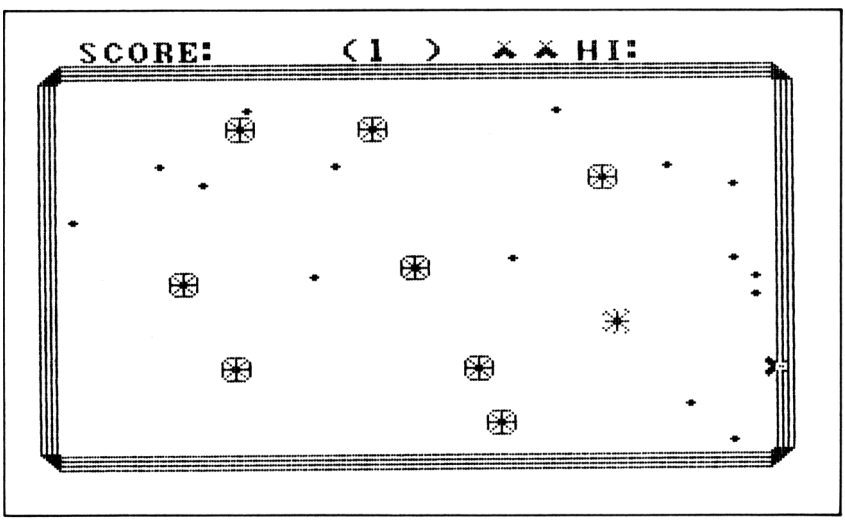
至於程式的執行，只要在BASIC 狀態鍵入LOAD，便可以了！請注意，程式必須執行無誤，完全除錯之後才可用POKE 82,213來SAVE

程式輯

大 蜘 蛛
外星人登陸(一)
分秒不差
PIANO
迷　　宮
摘 星 星
螺　　旋
外星人登陸(二)
精打細算
金 樸 克
節 目 單
創造自己的 GAME

第 7 章

大 蜘 蛛



遊戲說明

新生的小蝴蝶，對這個世界感到新鮮又好奇，快樂的在樹林間飛翔嬉遊……，可是背後恐怖的大蜘蛛，正瞪着可怕的眼睛準備捕捉它的食物呢！在這個遊戲裡，你可以用“A、Z”按鍵來令蝴蝶做上、下運動，而“←、→”則做左右右移動。畫面中央是大蜘蛛狩守的地點，許多分佈在畫面的小點則是蜘蛛的蛋，蝴蝶

每吃掉一個蛋便增加計分，當你的分數增加到某個數目，則便有獎分，並且畫面也自動換幕。遊戲時，要注意蜘蛛網的追擊，如果被網捕住，大蜘蛛便由網的中央出來將蝴蝶捕食。你總共有三隻蝴蝶，熟練了技術便可得高分，不過別忘記！四邊的框框也是蜘蛛網呢！

程式

```
]LIST

10  REM  ************************************
20  REM  **  SPIDER!!!SPIDER!!!SPIDER!!!  **
30  REM  ************************************
35  DIM RX(101),RY(101)
40  DIM TI(151)
45  DIM KI(220)
50  DIM IX(50),IY(50)
60  REM  ================
61  REM     POKE  SOUND
62  REM  ================
70  FOR I = 0 TO 22
71  READ WW
72  POKE 768 + I,WW
73  NEXT I
74  DATA 169,10,133,1,173,48
75  DATA 192,136,208,4,198,1
76  DATA  240,8,202,208,246
77  DATA 166,0,76,4,3,96
80  GOSUB 2000: CALL  - 144
85  GOSUB 500: GOSUB 600
90  GOSUB 700: GOSUB 800
```

```
100  REM ==========
101  REM  READ KEY
102  REM ==========
105 ZZ =  PEEK (P2)
110  IF KI(ZZ) = 0 GOTO 120
115 ZH = ZZ:R = RI(KI(ZH))
120  ROT= RO: XDRAW 3 AT X,Y
125  IF  PEEK (234) > 4 THEN  XDRAW 3 AT X,Y:
      GOSUB 425: GOTO 170
130 XN = X + XI(KI(ZH)):YN = Y + YI(KI(ZH))
135 X = XN:Y = YN
140  ROT= R: XDRAW 3 AT X,Y:RO = R
145 KK = X / T2 * T3 + XX + Y / T2
150  IF  PEEK (KK) = 1 THEN  GOSUB 400
170  GOSUB 200: GOTO 105
200  REM  --- TIME LOOP ---
210 TI(I) = TI(I) - 1: IF TI(I) < 0 THEN  GOSUB
    300
220  RETURN
300  REM  --- DRAW WEB ---
302  ON C1 GOSUB 310,320,330,330,330,350
304  CALL 768
306  RETURN
310 WX = X - IX(R):WY = Y - IY(R):C1 = C1 + 1

319  RETURN
320  POKE 769,30: CALL 768:C1 = C1 + 1: POKE
    769,1: RETURN
330  XDRAW 1 AT WX,WY: XDRAW 1 AT WX,WY:C1 =
    C1 + 1: RETURN
350  XDRAW 2 AT WX,WY:I = I + 1:C1 = 1
360  IF I > LV * 10 OR I > 150 THEN  GOSUB 47
    0
370  RETURN
400  REM  -- EGGSS EATEN --
405  FOR II = 1 TO 4: CALL 768: NEXT
```

```
410  ROT= 0: XDRAW 4 AT X,Y
415 EE = EE + 1
420 SC = SC + 10 * LV
422 EG = EG + 1: IF EG = 10 THEN  GOSUB 450:E
    G = 0
424  POKE KK,0: POKE 234,0: RETURN
425  REM  --SNAGGED ?--
426 B = X - 140:C = Y - 100:Q =  SQR (B * B +
    C * C)
427 UX = B / Q:UY = C / Q
428  POKE 769,200: CALL 768
430  FOR L = 1 TO Q STEP 2
432 XZ = 140 + L * UX:YZ = 100 + L * UY
434  XDRAW 1 AT XZ,YZ: XDRAW 1 AT XZ,YZ: NEXT

436  XDRAW 1 AT XZ,YZ: CALL 768
438  XDRAW 1 AT XZ,YZ: XDRAW 3 AT X,Y: IF WB =
    1 THEN WB = 0: DRAW 2 AT WX,WY
440 X = XO:Y = YO:SN = SN + 1: IF SN = 4 THEN
     GOTO 1100
442  ROT= 0: XDRAW 3 AT TX(SN),4
444  HCOLOR= 0: FOR II = 1 TO 19: HPLOT 128,1
    59 + II TO 144,159 + II: NEXT : XDRAW 3 AT
    X,Y:RO = 0: POKE P1,0:C1 = 1:R = 0:ZH =
    0
448  POKE 769,1: RETURN
450  REM  --SCORE --
452 CO = 0:SG$ =  STR$ (SR):LX = 65:LY = 10:
    GOSUB 1000
454 SR = SC
456 SG$ =  STR$ (SC)
458 CO = 3:LX = 65: GOSUB 1000
460  RETURN
470  REM  --LEVEL CHANGE --
471  FOR JJ = 1 TO 25:CO = 3 - CO: HCOLOR= CO
    : CALL 768
```

```
472  FOR II = 4 TO 8 STEP 2
473  HPLOT II,20 TO II,180: HPLOT 268 + II,20
      TO 268 + II,180: NEXT : NEXT
474  CALL 768: GOSUB 450:EG = 0
475  IF EE = 0 THEN  GOTO 480
476  FOR JJ = 1 TO EE: FOR LL = 1 TO 6: CALL
     768: NEXT :SC = SC + 10 * LV: GOSUB 450:
      NEXT
477  FOR II = 1 TO NE
478  POKE (RX(II) / 8 * T3 + XX + RY(II) / 8)
     ,0: NEXT
479 LV = LV + 1:EE = 0
480  HCOLOR= 0: FOR II = 19 TO 181: HPLOT 11,
     II TO 269,II: NEXT
484  REM  -- CHANGE LEVEL--
485 CO = 0:SG$ =  STR$ (LV - 1):LX = 125:LY =
     10: GOSUB 1000:CO = 3:SG$ =  STR$ (LV):L
     X = 125: GOSUB 1000
486 X = XO:Y = YO: ROT= 0:RO = 0:R = 0:ZH = 0
     : POKE P1,0: HCOLOR= 3
487  DRAW 2 AT 140,100: GOSUB 845
490  RETURN
500  REM  ============
501  REM   INITIALIZE
502  REM  ============
505  POKE 234,0: SPEED= 255
510 LV = 1:SN = 1
515 XX = 28500:T2 = 8:T3 = 34
520 P1 =  - 16368:P2 =  - 16384
530  POKE P1,0
535 XO = 136:YO = 168
540 X = XO:Y = YO:NM = 4:C1 = 1
545 L = 0:RI(0) = 0
550  IF EN = 1 THEN  GOTO 570
570  POKE 0,20: POKE 769,1
575  POKE 232,120: POKE 233,105
```

```
580  HGR2 : SCALE= 1: ROT= 0: HCOLOR= 3
585  IF  PEEK (800) = 0 THEN R =  RND ( -  PEEK
     (78) -  PEEK (79) * 256)
590  IF EN = 1 THEN  RETURN
595  GOSUB 599: RETURN
599  FOR I = 0 TO 1175: POKE XX + I,0: NEXT :
      RETURN
600  REM  =========
601  REM    DATA
602  REM  =========
615  DATA  193,149,136,218
620  DATA  0,-8,8,0,-8,0,0,8
622  DATA  3,20,11,12,269,12,277
624  DATA  20,277,180,269,188
626  DATA   11,188,3,180,3,20
628  DATA  0,16,48,32
630  DATA  0,-5,5,0,-5,0,0,5
635  DATA  158,173,188
640  DATA   " SCORE:","HI:"
645  DATA   " PRESS RETURN TO START"
646  DATA  "PRESS ESC TO END"
648  DATA  "(",")"
650  REM  -- READ ARRAYS --
655  FOR I = 1 TO 4: READ K:KI(K) = I: NEXT
658  FOR I = 1 TO 4: READ XI(I),YI(I): NEXT
660  FOR I = 1 TO 9: READ XA(I),YA(I): NEXT
662  FOR I = 1 TO 4: READ RI(I): NEXT
664  FOR I = 1 TO 4: READ J,K:IX(RI(I)) = J:I
     Y(RI(I)) = K: NEXT
670  FOR I = 1 TO 3: READ TX(I): NEXT
680  FOR I = 1 TO 6: READ WD$(I): NEXT
699  RETURN
700  REM  -------
701  REM  SETUP
702  REM  -------
705 CO = 3:I = 1:A = 2
```

```
710  FOR J = 1 TO 4
720  HPLOT XA(I),YA(I)
730  FOR I = 1 TO 9: HPLOT  TO XA(I),YA(I):
     NEXT
731  GOSUB 732:I = 1: NEXT : GOTO 750
732 XA(1) = XA(1) + A:YA(2) = YA(2) + A
734 YA(3) = YA(3) + A:XA(4) = XA(4) - A
736 XA(5) = XA(5) - A:YA(6) = YA(6) - A
738 YA(7) = YA(7) - A
739 XA(8) = XA(8) + A:XA(9) = XA(9) + A
740  RETURN
749  REM ---------
750 SG$ = WD$(1):LX = 10:LY = 10: GOSUB 1000
755 SG$ = WD$(5):LX = 115:LY = 10: GOSUB 1000

760 SG$ = WD$(6):LX = 145:LY = 10: GOSUB 1000

765 SG$ = WD$(2):LX = 200:LY = 10: GOSUB 1000

770  IF EN = 1 THEN HS =  PEEK (805) * 256 +
      PEEK (806):SG$ =  STR$ (HS):LX = 225:
      GOSUB  1000
775  DRAW 2 AT 140,100: DRAW 1 AT 140,100
799  RETURN
800  REM  -- START--
805 CO = 3: GOSUB 875
810 ZZ =  PEEK (P2)
815  IF ZZ = 141 THEN 830
820  IF ZZ = 155 THEN  GOTO 1150
825  GOTO 810
830 CO = 0: GOSUB 875
835  FOR I = 1 TO 3: XDRAW 3 AT TX(I),4: CALL
     768: NEXT
838  HCOLOR= 3: GOSUB 845:SG$ =  STR$ (LV):LX
      = 125:LY = 10: GOSUB 1000
840  XDRAW 3 AT TX(SN),4: RETURN
```

```
845 VV = 20 - LV * 5: IF VV < 0 THEN VV = 0
846 NE = LV * 20: IF NE > 100 THEN NE = 100
847 WL = LV * 10: IF WL > 150 THEN WL = 150
848  FOR I = 1 TO WL:TI(I) =  INT ( RND (1) *
     VV): NEXT
850  FOR I = 1 TO NE:RX(I) =  INT ( RND (1) *
     32) + 2:RY(I) =  INT ( RND (1) * 20) + 3

852  IF RX(I) > 15 AND RX(I) < 19 AND RY(I) <
     19 AND RY(I) < 23 THEN  GOTO 869
855 PV = RX(I) * T3 + XX + RY(I)
860  POKE PV,1
862 RX(I) = RX(I) * 8:RY(I) = RY(I) * 8
865  HCOLOR= 3: DRAW 4 AT RX(I),RY(I): CALL 7
     68: CALL 768
869  NEXT
872  XDRAW 3 AT X,Y
874 CO = 3: HCOLOR= 3:I = 1: RETURN
875  REM  ----- DISPLAY ----
880 SG$ = WD$(3):LX = 50:LY = 70: GOSUB 1000
885 SG$ = WD$(4):LX = 65:LY = 130: GOSUB 1000
890  RETURN
1000  REM  --- PRINT WORDS ---
1003  POKE 232,168: POKE 233,97
1005  HCOLOR= CO: ROT= 0
1010  FOR II = 1 TO  LEN (SG$)
1020 SV =  ASC ( MID$ (SG$,II,1))
1030  DRAW SV - 31 AT LX,LY
1040 LX = LX + 9: NEXT
1045  POKE 232,120: POKE 233,105
1050  RETURN
1100  REM  ----- END ----
1105  GOSUB 450
1110  IF SC > HS OR SC = HS THEN N1 =  INT (S
     C / 256):N2 = SC - N1 * 256: POKE 805,N1
     : POKE 806,N2
```

```
1115  GOSUB 599: CLEAR :EN = 1: GOTO 10
1150  IF  PEEK (8000) <  > 1 THEN  HGR2 : HOME
     : TEXT : HTAB 18: VTAB 10: PRINT "ENDED"
     : END
2000  REM ----AUTO READ----
2010  DIM Y$(50)
2020  HOME : INVERSE : VTAB 10: HTAB 11
2030  FOR D = 0 TO 1000: NEXT D: PRINT "WAIT
     !!! PLEASE !!!": NORMAL
2040 Y$ = "61A8.6FFFR N D823G"
2050  FOR I = 1 TO  LEN (Y$)
2060  POKE 511 + I, ASC ( MID$ (Y$,I,1)) + 12
     8
2070  NEXT I
2080  POKE 72,0
2090  RETURN
```

圖號	1	2	3	4
圖案				

圖案及其圖形號碼

```
*6978.69F8

6978- 04 00 0A 00 2E 00 63 00
6980- 7B 00 35 3F 24 2D 0C 0C
6988- 0C 15 96 E2 3F 16 0E 0E
6990- 1E DF 20 FC 1E 1E 0E 04
6998- 40 C0 39 BF 04 C0 08 0C
69A0- 0E 0E 0D 24 04 00 35 3F
```

```
69A8- 24 2D 0C 0C 0C 15 96 E2
69B0- 3F 16 0E 0E 1E DF 20 FC
69B8- 1E 1E 0E 04 40 C0 39 BF
69C0- 04 C0 08 0C 0E 0E 0D 24
69C8- 3C 3F 9F 12 36 16 B6 49
69D0- 2D 2D 0D 40 24 20 24 C0
69D8- 3B 3F 00 DB 40 15 0E 0C
69E0- 0C 96 3B BF 2D 2D 15 3F
69E8- 3F 3F 17 2D 6D 2D F5 FF
69F0- 1B 3F 00 3F 0C AD 1E 3F
69F8- 00
```

圖案圖形表

變數使用說明

ZZ ……按鍵的 ASCII 值，ZZ = PEEK(−16384)

ZH ……用來暫存 ZZ 的值

R ……旋轉值

RO ……上一次的旋轉值

X，Y ……上次畫蝴蝶的座標值

XN，YN ……目前所將畫蝴蝶的座標值

PV ……由亂數函數所造出的變數，用以設定蜘蛛蛋位置所對應的畫面記憶位址，若有蛋存在，則其對應位址內含值爲“1”。

KK ……由畫面位址取出的內含值，若其值爲 1 ，則該位置有一個蛋。

XX ……用來計算 KK 位址的基綫位址。

T2 ……計算位址用的常數(8)。

T3 ……計算位址用的常數，用以保證每一個蛋有其唯一對應的位址。

C1 ……畫蜘蛛網的分支變數。

WX，WY……畫蜘蛛網的座標位置。

I ……蜘蛛網的總個數。

LV ……畫面換幕的次數。

EE ……一幕中，所碰到蛋的總數。

EG ……吃蛋的計數，若到達 10 次，則顯示計分，並重新計數。

SC ……分數。

B，C ……大蜘蛛到獵物距離向量的垂直、水平分量。

Q……大蜘蛛到獵物的距離。

UX，UY……單位向量（步進值）。

WB ……蜘蛛網旗號。

XO，YO……蝴蝶的起始位置。

SN ……剩餘的蝴蝶數目。

CO ……顏色值。

LX，LY……字幕、字形位置。

SG$……表示分數的字串變數。

NE ……該幕中蛋的總數。

D$ ……CTRL － D ＋ RETURN

EN ……結束旗號。

HS ……最高分記錄。

VV ……計時迴圈的亂數值。

矩陣使用說明

KI(220) ……按鍵檢知矩陣，其足碼＊爲所按鍵的ASCII 值，按A、Z、←、→對應的KI 值爲1—4 。

XI(04),YI(04) ……X，Y的步進值，其足碼爲KI(ZZ)，因爲 KI 值爲1—4，因此 XI,YI 的維數爲 4 。

IX(50),IY(50) ……追踪蝴蝶位置的矩陣，用來畫蜘蛛網。

RX(101),RY(101)…以亂數來畫蛋的位置。

WDS(6)……字形用字串矩陣。

TI(151) ……計時迴圈的亂數。

遊戲鍵說明

RETURN ……遊戲開始。

ESC ……遊戲結束。

A ……蝴蝶向上移動。

Z ……蝴蝶向下移動。

← ……蝴蝶向左移動。

→ ……蝴蝶向右移動。

程式說明

10 — 50 行 ……宣告維數。

60 — 77 行 ……載入發聲副程式。

80 行 ……自動讀取圖形表（限磁帶使用者）。

85 — 90 行 ……至各副程式做初始設定。

100 — 170 行 ……讀鍵，消去蝴蝶舊位置，並且檢驗是否與蜘蛛網碰撞，若碰撞，則至碰撞處理副程式。若否，則計算蝴蝶新位置，然後在新位置畫出蝴蝶來，接着查看此新位置是否有蜘蛛蛋，若有則消掉，至延時程式。

200 — 20 行……計時程式，若 TI(I) < 0 則蜘蛛佈網。

300 — 370 行 ……織網副程式。

300 — 306 行 ……ON CI 至不同的織網副程式。

其中：

310 — 319 行 ……計算織網位置。

320 行 ……發聲程式。

330 行 ……造出蜘蛛網的閃爍效果。

350 — 370 行 ……織網。

400 — 424 行 ……吃蛋程式：發出聲音，將蛋由畫面上消去，更改分數及蛋的總數，將 0 Poke 到被消去的蛋之位址內。

425 — 448 行 ……計算由大蜘蛛到被捕獵物（網中心）的距離向量，令蜘蛛由沿向量移動去抓住獵物

，若是最後一隻蝴蝶，則結束遊戲，否則換幕開始。

450－466行 ……分數顯示程式。

470－480行 ……閃動左右畫框，更改分數，加獎分，蛋位址歸0，換幕。

485－490行 ……改變幕數，進入更高一幕。

500－599行 ……初值設定，若是磁碟機使用者，請在此加入560 D$=CHR$(13)+CHR$(4):PRINT D$;"BLOAD ALPHA－TBL"及561 PRINT D$;"BLOAD SPIDER－TBL"分別將字形及圖案圖形表載入。

600－699行 ……設定所有矩陣元素的初始值。

700－740行 ……畫面設定。

750－799行 ……畫出計分字形，並由805及806位址上取出最高分之記錄。

800－840行 ……等候按鍵。

845－874行 ……畫蜘蛛蛋於畫面上。

875－890行 ……畫出片頭文字。

1000－1115行……更改最高分記錄，將蛋位址歸0。

1150－1200行……程式結束，若使用第17章的節目單程式則會回到節目單上（限磁碟機用者）。

2000－2490行……自動讀造形表副程式（限錄音帶用者）。

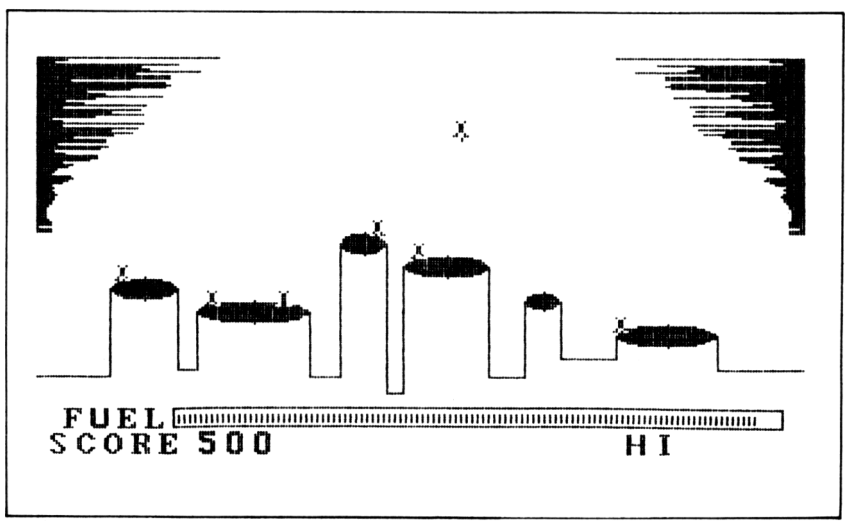
注意事項

上述的行號當中，使用磁碟機時，必須將 BLOAD 等指令加入程式當中，而將有關限磁帶用者之行號去除。程式當中的最高分存放位址（805，806）是第 3 頁區可由使用者自由使用的位址，其位置位於發聲程式之後。

至於圖形表部分，若爲磁碟機用者，則先前在第 3 章存入的字形表名稱必須是“ APLHA－TBL ”，而本章所列之圖案圖形表亦須以“　SPIDER-TBL　”的名稱存入磁片當中。若您使用的是磁帶，請將先前在第 3 章已錄好的字形表讀回到 APPLE 的 RAM ，然後鍵入本章所列之圖案圖形表，再鍵入主程式（但由 2000～2090 行的 AUTO READ 暫不鍵入程式便可執行。一旦程式證明無誤可以正確執行後，加入 2000～2090 行，再依第 6 章所說的方法，POKE 82，213然後將主程式存入磁帶，並緊接着 CALL－151，以 61A8.6FFFW 將圖形表錄至其後便可以了。

第 8 章

外星人登陸(一)



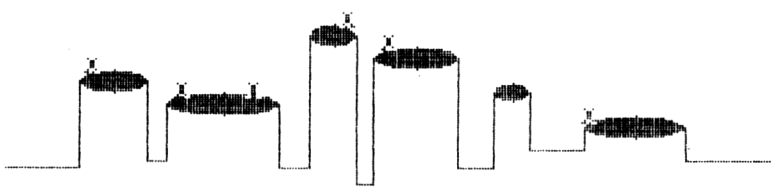
遊戲說明

太空登陸艇由螢幕的左上方出現，你必須小心的控制才可以令登陸艇安全着陸。移動搖桿，可以控制登陸艇噴射推進器的方向，推進器的方向共分左、右、下、左下、右下等五個方向，因此登陸艇是沿其反作用力方向受力而移動的，適當的控制搖桿便可以使登陸艇安全着陸。在陸地的六個高原上，如果登陸艇下降

速度太快，或碰到高原以外的地區，便算墜毀了。遊戲按鈕(0)是令推進器噴射的按鈕，當你以搖桿調整噴射火焰到所需要的方向後，必須按下第 0 號按鈕，噴射器才會噴射而有推進作用。着陸在面積較小的高原得分較高，反之則較低，而下降速度的快慢亦分三級：

尚可：基本分數
良好：基本分數× 2
極佳：基本分數× 3

每個高原的基本分數如下：



基本
分數： 100 50 150 100 200 100

程式

```
]LIST

10  REM  ************************
20  REM  **    ALIEN  LANDER    **
30  REM  ************************
42  REM
45  DIM VL(75),LX(100),LY(100)
47  REM
50  REM  ==============
51  REM     POKE SOUND
52  REM  ==============
53  FOR I = 0 TO 22
54  READ WW
55  POKE 768 + I,WW
56  NEXT I
57  DATA  169,10,133,1,173,48
58  DATA  192,136,208,4,198,1
59  DATA  240,8,202,208,246
60  DATA  166,0,76,4,3,96
65  REM  ================
70  REM         MAIN
80  REM  ================
90  GOSUB 500: GOSUB 600
95  GOSUB 700: GOTO 100
100  REM  ------------
101  REM  READ PADDLE
102  REM  ------------
103  FOR T = 1 TO 400: NEXT
105 PV =  PDL (0)
110 R1 =  INT (PV / PF)
115  IF  PEEK (P) > 127 AND FS = 1 THEN X
     I = XI + (PV - 128) / 256:YI = YI -
     (128 -  ABS (PV - 128)) / 512: GOSUB
     200
```

```
120 XA = XA + XI:YI = YI + .07:YA = YA +
    YI
125  IF YA < 10 THEN YI =  - YI:YA = 10
130  IF XA > 276 THEN XA = 1
135  IF XA < 1 THEN XA = 276
140  IF YA > 191 THEN  POKE 234,2: GOTO 1
    85
145  ROT= R2: XDRAW 2 AT XB,YB
150  ROT= R1: XDRAW 2 AT XA,YA
155  ROT= 0: XDRAW 1 AT XB,YB
160  ROT= 0: XDRAW 1 AT XA,YA
165 XB = XA:YB = YA:R2 = R1
170  REM  -- LAND CHEDK --
175 IX =  INT (XA / 4)
180  IF YA + 6 > VL(IX) THEN  IF YI < 3 THEN
     GOSUB 300: GOTO 195
185  IF  PEEK (234) > 1 AND  PEEK (234) <
    14 THEN  GOSUB 400
190  FOR T = 1 TO 50: NEXT
195  POKE 234,0: GOTO 105
200  REM  -- FUEL LOSS --
205 FU = FU - 1: IF FU = 0 THEN FU = 3:FX
     = FX - 1: HCOLOR= 0: HPLOT FX,FY TO
    FX,FY + 4: IF FX = 52 THEN FS = 0
210  RETURN
300  REM  -- LANDED --
302  IF YA - VL(IX) > 3 THEN  RETURN
303  ROT= R1: XDRAW 2 AT XA,YA: ROT= 0:CO
     = 3
305  FOR L = 1 TO 10: CALL 768: POKE 0,10
     * L: XDRAW 1 AT XA,YA: NEXT
310 N = N + 1:LX(N) = XA:LY(N) = YA
315  IF YI < 1 THEN WD = 4:LX = 110:LY =
    85: GOSUB 950: FOR T = 1 TO 40: NEXT
    :CO = 0:LX = 110: GOSUB 950: HCOLOR=
    3
```

```
320  GOSUB 350:XA = 10:YA = 10:XB = XA:YB
      = YA
325  ROT= 0
330  IF FS = 0 THEN  GOTO 1100
335  DRAW 1 AT XA,YA:R2 =  INT ( PDL (0) /
     PF): ROT= R2: DRAW 2 AT XA,YA
340 XI = 1:YI = 0
345  RETURN
350  REM  -- SCORE --
355 CO = 0:NX = 60:NY = 191: GOSUB 1000
358  IF YI < 0 THEN YI = 0
360  FOR L = 1 TO 6
365  IF XA < XO(L + 1) - A(L + 1) THEN SR
      = SV(L) * (3 -  INT (YI)):L = 6
370  NEXT
372 SC = SC + SR
375 CO = 3:NX = 60:NY = 191: GOSUB 1000
380  IF SC >  = 5000 AND BN = 0 THEN WD =
     7:LX = 110:LY = 85: GOSUB 950: CALL
     768: CALL 768:CO = 0:LX = 110: GOSUB
     950: GOSUB 742:FX = 269:BN = 1:FS =
     1
390  RETURN
400  REM  -- CRASH --
405  XDRAW 1 AT XA,YA: ROT= R1: XDRAW 2 AT
     XA,YA
410  ROT= 0: HCOLOR= 5
415  POKE 0,150
420  FOR L = 1 TO 10: XDRAW 3 AT XA,YA: CALL
     768: NEXT
425 CO = 0:NX = 60:NY = 191: GOSUB 1000
430 SC = SC - 100:CO = 3:NX = 60: GOSUB 1
     000
435 XA = 10:YA = 10:XB = XA:YB = YA
440  IF FS = 0 THEN  GOTO 1100
445  DRAW 1 AT XA,YA:R2 =  INT ( PDL (0) /
```

```
     PF): ROT= R2: DRAW 2 AT XA,YA
450 XI = 1:YI = 0
455  RETURN
500  REM  ---------------
501  REM    INITIALIZE
502  REM  ---------------
505 XA = 10:YA = 10
510 XB = XA:YB = YA
515 YF = 171:PF = 8
520 XI = 1:YI = 0: SPEED= 255
525 SP = 1:FS = 1:FU = 5:FX = 269:FY = YF
      + 3:XH = 0
530 P1 =  - 16368:P2 =  - 16384:P =  - 16
     287
535 D$ =  CHR$ (13) +  CHR$ (4)
540  FOR I = 0 TO 70:VL(I) = 160: NEXT
545  IF EN = 1 THEN  GOTO 565
550  PRINT D$;"BLOAD ALPHA-TBL"
560  PRINT D$;"BLOAD LANDER-TBL"
565  POKE 0,100: POKE 769,15
570  POKE 232,120: POKE 233,105
575  HGR2 : SCALE= 1: ROT= 0: HCOLOR= 3
599  RETURN
600  REM  ---------
601  REM  DATA
602  REM  -------
605  REM  --- WORDS --
610  DATA  "SCORE", "FUEL", "HI","PERFECT
     "
615  DATA  "PRESS RETURN TO START"
620  DATA  "PRESS ESC TO END"
625  DATA  "BONUS FUEL"
630  REM  -- LANDING PAD ---
635  DATA  40,120,12,5
640  DATA  80,130,20,5
642  DATA  120,100,8,5
```

```
644  DATA  150,110,15,5
646  DATA  185,125,6,4
648  DATA  230,140,18,5
650  DATA  279,0,0,0
652  REM  -- BOTTOM LINE--
658  DATA  158,155,158,165,158,150,155
660  REM  -- SCORE VAL
665  DATA  100,50,150,100 ,200,100
670  REM  -- READ ARRAYS--
675  FOR I = 1 TO 7: READ WD$(I): NEXT
680  FOR I = 1 TO 7: READ XO(I),YO(I),A(I
     ),B(I)
685  NEXT
690  FOR I = 1 TO 7: READ BL(I): NEXT
695  FOR I = 1 TO 6: READ SV(I): NEXT
699  RETURN
700  REM  -----
701  REM  SETUP
702  REM  -------
704 CO = 3
706  FOR I = 1 TO 77:J = ( RND (1) * (65 -
     I) + 5)
708  IF J < 0 THEN J = 5
710  HPLOT 0,18 + I TO J,18 + I
712  HPLOT 279,18 + I TO 279 - J,18 + I: NEXT

714  FOR I = 1 TO 6: GOSUB 800
716  FOR J = X1 + 3 TO X2 - 3:JJ =  INT (
     J / 4)
718 VL(JJ) = Y1: NEXT
720  NEXT : HPLOT XH,BL(I) TO 279,BL(I)
730  REM -SET UP FUEL/ ETC-
732 WD = 2:LX = 10:LY = 180: GOSUB 950
734 WD = 1:LX = 5:LY = 191: GOSUB 950
736 WD = 3:LX = 215: GOSUB 950
738  HPLOT 50,YF TO 270,YF TO 270,YF + 8 TO
```

```
     50,YF + 8 TO 50,YF
740  GOSUB 742: GOTO 748
742  HCOLOR= 6: FOR I = 1 TO 4
744  HPLOT 52,YF + I + 2 TO 267,YF + I +
     2: NEXT
746  HCOLOR= 3: RETURN
748  GOSUB 760: GOTO 799
760  REM  -- START--
761  POKE P1,0
762  GOSUB 765: GOTO 770
763  REM  -- DRAW WORDS--
765 WD = 5:LX = 30:LY = 62: GOSUB 950
768 WD = 6:LX = 55:LY = 80: GOSUB 950: RETURN

769  REM  -- LOOP--
770 ZZ =  PEEK (P2)
772  IF ZZ = 155 THEN  GOTO 1150
774  IF ZZ <  > 141 THEN  GOTO 770
780 CO = 0: GOSUB 765: HCOLOR= 3
785 R2 =  INT ( PDL (0) / PF): ROT= R2: DRAW
     2 AT XA,YA
790  ROT= 0: DRAW 1 AT XA,YA
795  RETURN
799  RETURN
800  REM  -COMPUTE&DRAW ELLITSE-
805 X1 = XO(I) - A(I) - 1:Y1 = YO(I):X2 =
     XO(I) + A(I)
810  HCOLOR= CO
815 YS = YO(I) - B(I):YE = YO(I) + B(I)
820  FOR Y = YS TO YE STEP SP
825 YY = Y - YO(I)
830 XX = A(I) *  SQR (1 - (YY * YY) / (B(
     I) * B(I)))
835 XS = XO(I) - XX:XE = XO(I) + XX
840  FOR X = XS TO XE STEP SP
845  HPLOT X,Y: NEXT : NEXT
```

```
879  REM  -DRAW LOWER TERRAIN-
880  HPLOT X1,Y1 TO X1,BL(I): HPLOT X2,Y1
      TO X2,BL(I + 1): HPLOT XH,BL(I) TO
     X1,BL(I):XH = X2
890  RETURN
950  REM -- PRINT WORDS--
955  HCOLOR= CO: POKE 232,168: POKE 233,9
     7
960  FOR II = 1 TO  LEN (WD$(WD))
965 SV =  ASC ( MID$ (WD$(WD),II,1))
970  DRAW SV - 31 AT LX,LY:LX = LX + 10: NEXT

975  POKE 232,120: POKE 233,105
980  RETURN
1000  REM -- SCORE--
1010  HCOLOR= CO: POKE 232,168: POKE 233,
     97
1020 SC$ =  STR$ (SC)
1030  FOR II = 1 TO  LEN (SC$):SV =  ASC
     ( MID$ (SC$,II,1))
1040  DRAW SV - 31 AT NX,NY:NX = NX + 10:
      NEXT
1045  POKE 232,120: POKE 233,105
1048  IF SC <  - 5000 THEN  GOTO 1100
1050  RETURN
1100  REM  --END--
1102  ROT= 0
1104  FOR I = 1 TO 10: POKE 0,15 * I: CALL
     768
1108  NEXT
1110  POP
1115  IF SC <  = HS THEN  GOTO 1139
1120 SH = SC:SC = HS:CO = 0:NX = 240:NY =
     191: GOSUB 1000
1130 SC = SH:CO = 3:NX = 240: GOSUB 1000
1135 HS = SC
```

```
1139  REM  -ANOTHER GAME ?-
1140  GOSUB 760:FX = 269: GOSUB 742
1141 CO = 0:NX = 60: GOSUB 1000:SC = 0
1142  IF N = 0 THEN  GOTO 1148
1144  FOR I = 1 TO N
1146  ROT= 0: XDRAW 1 AT LX(I),LY(I): CALL
      768: NEXT :N = 0
1148 FS = 1:XI = 1:YI = 0: GOTO 100
1150 IF PEEK(800) <  > 1 THEN  HGR2 : HOME
      : TEXT : HTAB 18: VTAB 10: PRINT "EN
      DED": END
1160  PRINT D$;"RUN MENU"
1200  REM ============
```

圖號	1	2	3
圖案			

圖案及其圖形號碼

```
*6978.69A2

6978- 03 00 08 00 17 00 1C 00
6980- 24 07 E0 8D 05 A0 D2 33
6988- F6 1E 6D 29 1C 04 00 49
6990- ED A3 06 00 2D 38 3F 1C
6998- 2D 2D B5 3A 3F 3F 0E 2D
69A0- F5 3F 00
```

圖案圖形表

變數使用說明

T……迴圈變數。

PV ……搖桿函數值（PDL(0)＝ 0～255）。

P_1 ……新的旋轉值。

P_2 ……舊的旋轉值。

PF ……搖桿函數調整因子，用以將搖桿函數值區分成對應的旋轉值。

P……按鈕位址（－16287），其內含值若大於127則表示按鈕被按下。

FS ……油量旗號，FS ＝ 0 表示油已耗盡。

XA、YA ……登陸艇的座標。

XI、YI ……速度變數，以達成噴射器與地心引力的加速效果。

XB、YB ……暫存前次畫圖的座標位置。

IX ……登陸艇的水平位置記錄變數，用以判斷是否成功的登上高原位置。

FU ……油量計數器。

FX、FY ……畫油量表的座標變數。

N……登陸艇計數器。

BN ……獎分旗號，當 BN ＝ 0時，則增加額外油料。

CO ……高解像圖形彩色值。

SR ……分數增量。

SC ……分數。

HS ……最高分記錄。

SH ……分數暫存。

LX、LY ……文字字形顯示的座標位置。

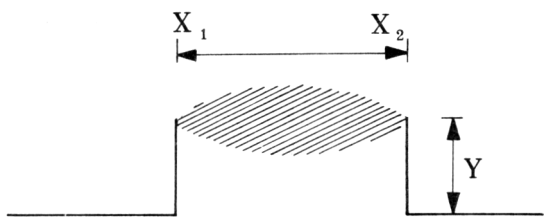
NX、NY ……分數字形的座標位置。

D$ …… CTRL－D＋ RETURN （限磁碟機使用者）。

WD ……字串矩陣變數 WD$ 的足碼。

YF ……畫油量計的圖框。

XI、YI、X2…高原的高度及邊界座標。



XS, XE
YS, YE ……畫高原頂端橢圓區的座標。
X, Y

SV ……圖號變數。

矩陣使用說明

LX(100), LY(100)…每個已登陸者的座標位置。

SV(6) ……每一高原的基本分數。

BL(7) ……谷底垂直座標。

XO(7), YO(7) ……高原頂端的橢圓形中心座標。

A(7), B(7)……每一橢圓的長軸及短軸。

遊戲鍵說明

RETURN ……開始遊戲。

ESC ……結束遊戲。

搖桿（0）……移動推進器噴射方向。

按鈕（0）……令推進器發射。

程式說明

10—55 行 ……宣告總數。

60—95 行 ……至各副程式做初始設定。

100—165 行 ……延時，讀搖桿值以設定推進器噴射方向。計算登陸艇位置，在舊位置消去登陸艇，在新位置畫出新登陸艇。

170—195 行 ……檢查是否在正確位置登陸，登陸速度是否適當。若一切正常，則登陸成功，否則檢知是否碰撞，若是則發聲並摧毀。

200—210 行 ……減少油量。

300—345 行 ……登陸艇已達到登陸位置，但是否由底下往上到達的呢？若是，則不算成功。若由上方登陸，則發聲、加分、並於 LX(N)，LY(N) 記錄登陸位置。

350—390 行 ……計分程式，並檢知是否超過 5000 分。

400—450 行 ……摧毀登陸艇的副程式，減分，並檢知油量

，若油量未完，則開始另一次登陸。

500—599行 …… 初始變數設定，550及560行是磁碟機使用者用來取圖形表的指令，若是錄音機使用者，請將此二行去掉。

600—699行 …… 矩陣元素設定。

700—720行 …… 畫出遊戲的場景。

730—746行 …… 畫出油量表。

748—799行 …… 畫出序幕文字，等候按鍵。

800—845行 …… 畫橢圓的副程式，第850行是橢圓的公式。

880—890行 …… 畫谷底副程式。

950—980行 …… 顯示文字的副程式。

1000—1050行…… 分數顯示。

1100—1135行…… 結束的發聲程式，並且更改最高分記錄。

1140—1148行…… 是否再玩一次？

1150—1200行…… 若是磁碟機使用者，則可以回到第19章的節目單程式，若是磁帶的使用者，則此段的程式不用。

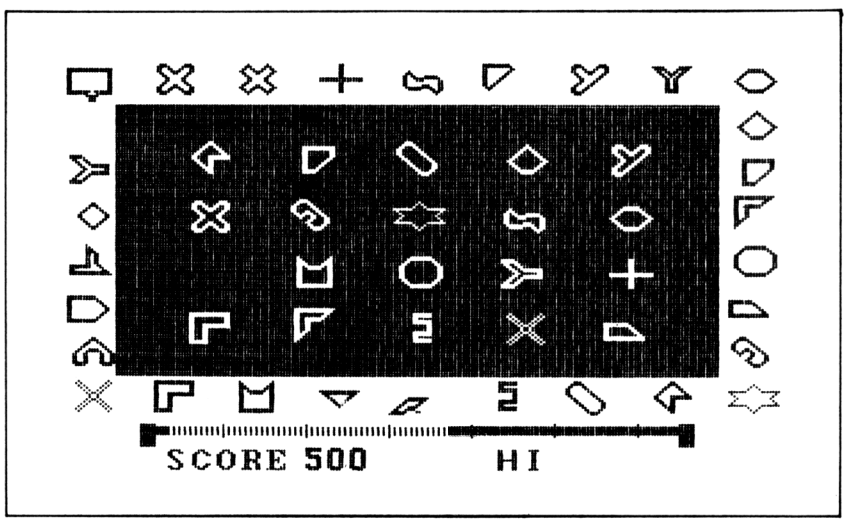
注意事項

本程式是寫成磁碟機使用的形態，如果您使用的是磁帶，請仿第7章的大蜘蛛或第6章的方法，將第535、550、560、1160等有關磁碟機指令去掉，同時將第7章大蜘蛛程式中的第2000～2090行附加到本程式末，最後加入： 48 GOSUB 2000 ：CALL—144便可以造出磁帶者用的程式了。當然

2000副程式、主程式及圖形表均必須測試檢查無誤才可以存到磁帶，同時別忘了POKE　82，213。

第 9 章

分秒不差



遊戲說明

在這遊戲中，你主要的目的是將框外的圖形精確的重疊在框內的圖形上。遊戲的一開始，畫面將會顯示框框，然後你可以按入 RETURN 鍵使遊戲開始，會有 30個圖形顯示在框子的外面，而其中的任何 20 個，也將會顯示在框子的裏面。你可以用 RETURN 及 ESC 鍵去控制圖形的移動，盡你的能力在限定時

間內把圖形重疊完疊完畢，在框子的下面會有圖形顯示所剩的時間，假如你在時間內完成所有圖形，那麼你將會得到再玩一次的機會而且分數會增加，其最高分數是 10000 分。如果要使圖形做更快的移動可以再按一次 RETURN 或 ESC ，其動作次序是正向→快→停→反向→快→……。

程式

```
]LIST

10  REM  ************************
20  REM  **    MATCHER  !!!    **
25  REM  ************************
30  REM
31  GOSUB 2000: CALL  - 144
32  REM  ==================
34  REM    POKE  SOUND
36  REM  ==================
38  FOR I = 0 TO 22
40  READ WW
42  POKE 768 + I,WW
44  NEXT I
45  DATA  169,10,  133,1,173,48
46  DATA  192,136,208,4,198,1
47  DATA  240,8,202,208,246
48  DATA  166,0,76,4,3,96
50  DIM SH(31),P(31),SM(31)
55  DIM IX(21),IY(21)
60  DIM OX(31),OY(31)
70  DIM X(10),Y(10)
```

```
80  REM   --------------
85  REM    MAIN ROUTINE
90  REM   --------------
95  GOSUB 700: GOSUB 800: GOSUB 900
100  REM  ----------
101  REM    INPUT
102  REM  ----------
125 ZZ =  PEEK (P2)
130  GOSUB 490
135  IF ZZ = 160 THEN  GOSUB 500
140  IF ZZ = 141 THEN S1 = S1 + 1:S2 = 0: POKE
     P1,0
145  IF ZZ = 155 THEN S2 = S2 + 1:S1 = 0: POKE
     P1,0
150  GOSUB 200: GOTO 125
200  REM  -----------
201  REM   PLOT DIR
202  REM  -----------
210  IF S1 > 0 THEN  GOSUB 400: ON S1 GOSUB
     240,242,244
220  IF S2 > 0 THEN  GOSUB 425: ON S2 GOSUB
     262,264,266
230  RETURN
240 A = A1: GOSUB 250: RETURN
242 A = SF * A1: GOSUB 250: RETURN
244 S1 = 0:A1 = A1 *  - 1: RETURN
250 SX = SX + A
252  XDRAW SH(I) AT SX - A,SY: XDRAW SH(I)
      AT SX,SY
254  RETURN
262 B = B1: GOSUB 300: RETURN
264 B = SF * B1: GOSUB 300: RETURN
266 S2 = 0:B1 = B1 *  - 1: RETURN
300 SY = SY + B
310  XDRAW SH(I) AT SX,SY - B: XDRAW SH(I)
      AT SX,SY
```

```
315  RETURN
400  REM  -- BOUNDRY  CHECK X --
405 SW = 1
410 BN = SX + A: IF BN < 9 OR BN > 271 THEN
     A1 = A1 *  - 1
420  GOSUB 450: RETURN
425  REM  -- BOUNDRY CHECK Y --
430 SW = 1
435 BN = SY + B: IF BN < 9 OR BN > 165 THEN
     B1 = B1 *  - 1
440  GOSUB 460: RETURN
450  REM   -- FIT CHTURN
455  IF SY = IY(J) THEN  IF  ABS (SX - IX(
     J)) <  ABS (A) THEN  XDRAW SH(I) AT S
     X,SY:SY = IY(J): XDRAW SH(I) AT SX,SY
     : GOSUB 470
459  RETURN
460  REM   -- FIT CHECK Y --
465  IF SX = IX(J) THEN  IF  ABS (SY - IY(
     J)) <  ABS (B) THEN  XDRAW SH(I) AT S
     X,SY:SY = IY(J): XDRAW SH(I) AT SX,SY
     : GOSUB 470
469  RETURN
470  REM  -- MATCH --
472  POKE P1,0:CT = CT + 1
476  POKE 0,25: POKE 769,20: FOR II = 1 TO
     10: POKE 0,25 + I: CALL 768: NEXT : POKE
     0,1: POKE 769,1:S1 = 0:S2 = 0:SM(I) =
     1
478  GOSUB 485
480  IF CT = SN THEN  GOTO 1200
482 SW = 0: GOSUB 500
484  RETURN
485 CO = 0:LX = 100:LY = 190: GOSUB 1150
486 SC = SC +  INT (TI / 30) * 100
487 CO = 3:LX = 100: GOSUB 1150: RETURN
```

```
489  RETURN
490  REM  -- TIME CHECK --
492  CALL 768:TC = TC - 1: IF TC > 0 THEN
      RETURN
494  HPLOT TI,172 TO TI,175:TI = TI - 1
496  IF TI = 40 THEN  GOTO 1200
499 TC = 10: RETURN
500  REM  ------------
501  REM  OUTTER MOVE
502  REM  ------------
510  POKE P1,0
520  IF SW = 1 THEN  XDRAW SH(I) AT SX,SY:
      XDRAW SH(I) AT OX(I),OY(I):SW = 0:S1
      = 0:S2 = 0
525 I = I + 1
530  XDRAW 31 AT OX(I - 1),OY(I - 1)
535  IF I = 31 THEN I = 1
540  GOSUB 580
545 J = P(SH(I))
550  XDRAW 31 AT OX(I),OY(I)
560 SX = OX(I):SY = OY(I)
570  CALL 768
575  RETURN
580  REM  --- OUTTER SKIP ----
585  IF I = 31 THEN I = 1
590  IF SM(I) = 1 THEN I = I + 1:CO TO 585

595  RETURN
700  REM
701  REM      INIT
702  REM  ----------
705 SN = 20:I1 = 10:I2 = 190
710 R = 30:TI = 230:TC = 0
715 SF = 8:Z = 0
720 M =  - 1:SI =  - 1
725 SE = 1:J = 1:CY = 20
```

```
730 XS = 65:YS = 60
735 XO = 22:YO = 145
740 SX = XO:SY = YO
745 A1 = 1:B1 = 1:S1 = 0:S2 = 0
750 A = 1:B = 1: SPEED= 255
755 P1 =  - 16368:P2 =  - 16384
762  DEF  FN RD(RL) =  INT ( RND (1) * RL)

764  IF  PEEK (800) <  > 1 THEN R =  RND (
      -  PEEK (78) -  PEEK (79) * 256)
766  FOR I = 1 TO 30:SH(I) = I: NEXT
768  GOSUB 790
770  HGR2
778  POKE 232,168: POKE 233,97
780  POKE 0,1: POKE 769,1
782  POKE P1,0
784  SCALE= 1: HCOLOR= 3: ROT= 0
789  RETURN
790  FOR I = 1 TO 30:SM(I) = 0:P(I) = 0: NEXT

799  RETURN
800  REM  ---------
801  REM    DATA
802  REM  ---------
805  REM  -- COORDINATES --
810  DATA    31,31,249,31,40,40,240,40
815  DATA   45,45,235,45,31,149,40,144
820  DATA   45,139
825  REM   --WORDS --
830  DATA   "PUSH*RETURN*TO*START*ESC*TO*E
     ND"
835  DATA   "SCORE","HI"
840  FOR I = 1 TO 9: READ X(I),Y(I): NEXT

845  FOR I = 1 TO 3: READ WD$(I): NEXT
850  REM  --- INIT SHAPE POS ----
```

```
852  FOR I = 1 TO 30
854 OX(I) = XO + C1:OY(I) = YO + C2
856  IF I = 7 THEN CY = 0:SI = 1:CX = 30
858  IF I = 15 THEN CX = 0:CY = 20
859  IF I = 22 THEN CY = 0:CX = 30:SI =  -
     1
860 C1 = C1 + SI * CX
862 C2 = C2 + SI * CY: NEXT
864 C1 = 0:C2 = 0:J = 1
866  FOR I = 1 TO 20
868 IX(I) = XS + C1:IY(I) = YS + C2
870 C1 = C1 + 38
875  IF J = 5 THEN C1 = 0:C2 = C2 + 25:J =
     0
880 J = J + 1: NEXT I
899  RETURN
900  REM  ----------
901  REM      SETUP
902  REM  ----------
905  FOR Y = Y(1) TO Y(7)
910  IF Y > Y(3) AND Y < Y(9) THEN  HPLOT
     X(5),Y TO X(6),Y
915  HPLOT X(1),Y TO X(2),Y
920  IF Y >  = Y(5) AND Y < Y(9) THEN  HPLOT
     X(5),Y TO X(6),Y
925  HPLOT X(1),Y TO X(3),Y: HPLOT X(4),Y TO
     X(2),Y
930  NEXT
935 CO = 3:WD = 1:LX = 2:LY = 190: GOSUB 1
     100
940  GOSUB 970: GOSUB 978
945 CO = 0:WD = 1:LX = 2: GOSUB 1100
950 CO = 3:WD = 2:LX = 50:LY = 190: GOSUB
     1100
955 LX = 170:WD = 3: GOSUB 1100
960  GOSUB 1000
```

```
965  RETURN
970  REM     ---- BEGIN ----
972  FOR I = 1 TO 10
973  HPLOT X(3),170 + I TO X(5),170 + I: HPLOT
     X(6),170 + I TO X(4),170 + I
974  NEXT
975  FOR I = 1 TO 5
976  HPLOT X(5) + I,172 TO X(5) + I,175: HPLOT
     X(6) - I,172 TO X(6) - I,175: NEXT
977  RETURN
978  POKE P1,0
979 ZZ =  PEEK (P2)
980  IF ZZ = 141 THEN  POKE P1,0: GOSUB 98
     5: GOSUB 990: GOTO 984
981  IF ZZ = 155 THEN  GOTO 1285
983  GOTO 979
984  RETURN
985  REM   ---- SET TIME ----
986  FOR I = I1 TO I2 STEP 2: CALL 768
987  HPLOT X(3) + I,172 TO X(3) + I,175
988  NEXT : POKE P1,0:I = 0
989  RETURN
990  FOR I = 30 TO 180 STEP 30
992  HPLOT X(3) + I,171 TO X(3) + I,176: NEXT
     : RETURN
1000  REM  --------------
1001  REM  RANDOM SHUFFLE
1002  REM  --------------
1005  GOSUB 1040: GOSUB 1060: GOSUB 1080: GOSUB
     1040: GOSUB 1090: RETURN
1040  REM  -- SHUFFLLE --
1045  FOR I = 30 TO 2 STEP  - 1
1050 J =  FN RD(I) + 1:T = SH(I)
1055 SH(I) = SH(J):SH(J) = T
1058  NEXT
1059  RETURN
```

```
1060  REM  -- SELECT 20 --
1065  FOR I = 1 TO 20:P(SH(I)) = I: NEXT
1070  RETURN
1080  FOR I = 1 TO 20: XDRAW SH(I) AT IX(I
      ),IY(I): NEXT
1089  RETURN
1090  REM  -- DRAW OUTTER --
1092  HCOLOR= 3
1094  FOR I = 1 TO 30: DRAW SH(I) AT OX(I)
      ,OY(I): NEXT
1096 I = 1: XDRAW 31 AT OX(I),OY(I)
1098 J = P(SH(I))
1099  RETURN
1100  REM  --- PRINT WORDS ----
1105  POKE 232,168: POKE 233,97
1110  HCOLOR= CO
1115  FOR II = 1 TO  LEN (WD$(WD))
1120 SV =  ASC ( MID$ (WD$(WD),II,1))
1125  DRAW SV - 31 AT LX,LY
1130 LX = LX + 9
1135  NEXT
1140  POKE 232,120: POKE 233,105
1145  RETURN
1150  REM  -- SCORE --
1155  POKE 232,168: POKE 233,97
1160  HCOLOR= CO
1165 SC$ =  STR$ (SC)
1170  FOR II = 1 TO  LEN (SC$)
1175 SV =  ASC ( MID$ (SC$,II,1))
1180  DRAW SV - 31 AT LX,LY
1185 LX = LX + 8: NEXT
1190  POKE 232,120: POKE 233,105
1199  RETURN
1200  REM  -----------
1201  REM      ENDING
1202  REM  -----------
```

```
1210  POP
1225  IF SN <  > CT THEN  POKE 0,240: POKE
      769,200: CALL 768: XDRAW SH(I) AT SX,
      SY: GOTO 1270
1250  REM  -- PERFECT --
1252 K = I: POKE 769,1
1254  FOR I = 1 TO 20
1256 RA =  INT ( RND (SE) * 20) + 1
1258  POKE 0,10 * RA: CALL 768
1260  XDRAW SH(RA) AT IX(RA),IY(RA)
1262  FOR T = 1 TO 50: NEXT
1264  XDRAW SH(RA) AT IX(RA),IY(RA):NEXY]1
      264 XDRAW SH(RA) AT IX(RA),IY(RA):NEX
      YIX(RA),IY(RA): NEXT
1266  FOR T = 1 TO 10: GOSUB 485: POKE 769
      ,15: CALL 768: NEXT
1268 I = K
1270  REM  --- PLAY AGAIN ---
1271  IF SC > HS THEN SH = SC:SC = HS:CO =
      0:LX = 190:LY = 190: GOSUB 1150:SC =
      SH:CO = 3:LX = 190: GOSUB 1150:HS = S
      C
1272  POKE P1,0
1274 ZZ =  PEEK (P2)
1280  IF ZZ = 141 THEN  GOTO 1290
1282  IF ZZ = 155 THEN  GOTO 1285
1284  GOTO 1274
1285  IF  PEEK (800) <  > 1 THEN  HGR2 : HOME
      : TEXT : HTAB 18: VTAB 10: PRINT "END
      EN": END
1286  REM
1290  REM  --- RESET ---
1291  XDRAW 31 AT OX(I),OY(I)
1292  HCOLOR= 0
1293  FOR I = 1 TO 30: DRAW SH(I) AT OX(I)
      ,OY(I)
```

```
1294  IF P(SH(I)) = 0 THEN  GOTO 1296
1295  HCOLOR= 3:J = P(SH(I)): DRAW SH(I) AT
      IX(J),IY(J): HCOLOR= 0
1296  NEXT : POKE 0,1: POKE 769,1
1297  HCOLOR= 0:I1 = 11:I2 = 189: GOSUB 98
      5: GOSUB 790:TI = 230
1298 CO = 0:LX = 100:LY = 190: GOSUB 1150:
      HCOLOR= 3: GOSUB 1000:SC = 0
1299 SX = XO:SY = YO:S1 = 0:S2 = 0:CT = 0:
      GOTO 100
1999  END
2000  REM     ==================
2010  REM         AUTO-READ!!
2020  REM     ==================
2030  DIM Y$(50)
2040  HOME : INVERSE : VTAB 10: HTAB 11
2050  FOR D = 0 TO 1000: NEXT D: PRINT "WA
      IT !!! PLEASE !!!": NORMAL
2060 Y$ = "61A8.6FFFR N D823G"
2070  FOR I = 1 TO  LEN (Y$)
2080  POKE 511 + I, ASC ( MID$ (Y$,I,1)) +
      128
2090  NEXT I
2100  POKE 72,0
2200  RETURN
```

圖號	1	2	3	4	5	6	7
圖案							

圖號	8	9	10	11	12	13	14
圖案							

圖號	15	16	17	18	19	20	21
圖案							

圖號	22	23	24	25	26	27	28
圖案							

圖號	29	30	31
圖案			

圖案及其圖形號碼

```
*6978.6ECE

6978- 1F 00 40 00 68 00 85 00
6980- B3 00 E6 00 23 01 48 01
6988- 6F 01 90 01 C4 01 F1 01
6990- 26 02 51 02 79 02 9F 02
6998- C4 02 EC 02 14 03 47 03
69A0- 7D 03 9E 03 CB 03 FE 03
```

```
69A8- 23 04 55 04 8A 04 C0 04
69B0- DC 04 0C 05 2A 05 4D 05
69B8- 3F 3F 24 24 24 24 24 2C
69C0- 2D 2D 2D 2D 2D 36 36 1E
69C8- 1E 1E 1E 1E 1E 3F 3F 24
69D0- 24 24 24 2C 2D 2D 2D 2D
69D8- 36 F6 1E 1E 1E 1E 06 00
69E0- 27 27 27 27 3F 28 2C 2C
69E8- 2C 2C 2C 2C AC 2E 2E 2E
69F0- 2E 2E 0E AF 3B 3E 3E 3E
69F8- 3F 77 2D 05 00 3F 07 E0
6A00- 1C 1C 1C 0C 0C 0C 0C 0C
6A08- 2D 2D AD 0E 0E 0E 0E 1E
6A10- 1E 1E 1E 1E 3F 3C 3F 1C
6A18- 1C 1C 1C 0C 0C 0C 0C 2D
6A20- 2D 75 0E 0E 0E 1E 1E 1E
6A28- 1E 3F 00 3F 3F 38 38 38
6A30- 20 24 24 0C 0C 0C 0C 2D
6A38- 2D 2D AD 15 15 15 36 36
6A40- F6 1E 1E 1E 3F 27 3F 3F
6A48- 1C 1C 1C 24 24 64 0C 0C
6A50- 2D 2D 2D 75 0E 0E 36 36
6A58- F6 1E 1E 3F 07 00 DB 3F
6A60- 3F 20 24 64 0C 0C 0C 0C
6A68- 0C 0C AD 15 15 15 15 15
6A70- 15 36 36 1E 3F 3F 20 24
6A78- 24 3F 3F 37 36 36 3F 3F
6A80- 24 24 0C 0C 0C 0C 0C 0C
6A88- 75 0E 0E 0E 0E 0E 36 36
6A90- 3F 27 24 24 D8 3F FF 12
6A98- 36 36 00 1C 1C 1C 3F 3F
6AA0- 3F 0C 0C 0C 07 E0 1C 2D
6AA8- 2D 2D 0C 0C 0C 15 0E 0E
6AB0- 2D 2D 2D 1E 1E 1E 15 0E
6AB8- 0E 3F 3F 3F 1E 1E 06 00
6AC0- 07 38 38 38 38 20 0C 0C
6AC8- 0C 0C 0C AD 0E 0E 0E 0E
6AD0- F6 1E 1E 1E 1E 3C 1C 1C
6AD8- 1C 1C 64 0C 0C 0C 75 0E
6AE0- 0E 0E F6 1E 1E 06 00 DB
6AE8- DB 20 27 27 25 27 25 27
6AF0- 25 27 25 27 2D 2E 2C 2E
6AF8- 2C 2E 2C 2E AC 15 37 37
6B00- 37 37 37 37 37 37 07 00
6B08- 07 38 38 38 38 38 38 38
```

```
6B10- 20 64 0C 0C 2D 15 15 15
6B18- 15 15 15 15 15 15 36 1E
6B20- 1E 1E 3F E4 1C 1C 1C 1C
6B28- 1C 1C 1C 24 0C 0C 2D 0E
6B30- 0E 0E 0E 0E 0E 0E 0E 36
6B38- 1E 1E 3F 00 01 C0 30 37
6B40- 37 37 07 20 27 67 25 25
6B48- 25 27 27 27 67 25 25 15
6B50- 2E 2E 2E 2C 2C 2C AC 2E
6B58- 2E 1E 37 37 37 35 35 35
6B60- F5 37 37 07 20 27 27 07
6B68- 00 01 C0 30 37 37 1E 3C
6B70- 3E 3C 2C 3C 0C 25 25 25
6B78- 27 27 27 27 25 67 35 25
6B80- 15 2E 2E 2E 2C 2C 64 2E
6B88- 2C 15 37 35 37 37 37 37
6B90- 35 35 35 35 77 37 27 37
6B98- 07 20 27 27 04 00 01 C0
6BA0- 30 37 37 1E 3C 3E 3C 2C
6BA8- 3C 0C 25 25 25 27 27 27
6BB0- 27 25 67 35 25 15 2E 2E
6BB8- 2E 2C 2C 64 2E 2C 15 37
6BC0- 35 37 37 37 37 37 37 06
6BC8- 00 3E 3C 3E 3C 3E 3C 3E
6BD0- BC 27 25 27 25 27 25 27
6BD8- 25 27 25 27 2D 2E 2C 2E
6BE0- 2C 2E 2C 2E 2C 2E 2E 2E
6BE8- 2E 2E 3E 3E 3E 3E 3E 36
6BF0- 00 01 C0 1C 1C 1C 1C 1C
6BF8- 1C 1C 2D 2D 2D 2D 2D 2D
6C00- 2D 1E 1E 1E 1E 1E 1E E7
6C08- 1C 1C 1C 1C 2D 2D 2D 2D
6C10- 2D 1E 1E 1E 1E 07 00 3C
6C18- 3C 3C 0C 25 25 27 27 37
6C20- 37 37 07 20 27 27 25 67
6C28- 25 25 35 25 15 2E 2E 2E
6C30- 2E 2E 2E 2E 3E 2E 1E 37
6C38- 37 27 04 00 01 C0 3C 3E
6C40- BC 3E 3E 3E 20 27 25 27
6C48- 25 67 25 35 25 35 25 15
6C50- 2E 2C 2E 20 25 25 15 2E
6C58- 3E 2E 3E 2E 1E 3F 3F 37
6C60- 2D 2D 05 00 01 C0 2C 2E
6C68- AC 2E 2E 2E 20 25 27 25
6C70- 27 E5 27 37 27 37 27 17
```

```
6C78- 3E 3C 3E 20 27 27 17 3E
6C80- 2E 3E 2E 3E 0E 2D 2D 35
6C88- 3F 3F 07 00 3E 3C 3E 3C
6C90- 3E 3C 3E BC 27 25 27 25
6C98- 27 25 27 25 27 25 27 2D
6CA0- 2E 2C 2E 2C 2E 2C 2E 2C
6CA8- 2E 2C 2E 2C 2E 2C 36 37
6CB0- 35 37 35 37 35 37 35 3F
6CB8- 3F 3F 2E 2D 2D 2D 00 DB
6CC0- DB 23 24 24 24 24 24 24
6CC8- 2D 2D 2D 2D 2D 2D 2D 36
6CD0- 36 36 3F 3F 3F 3F 36 36
6CD8- 36 3E 3F 3F 24 24 24 24
6CE0- 24 24 2D 2D 2D 2D 2D 2D
6CE8- 36 36 3F 3F 3F 3F 36 36
6CF0- 36 3E 3F 07 00 3C 3C 3C
6CF8- 3C 3C 3C 2C 2C 2C 2C 2C
6D00- 2C 2C 2E 2E 2E 2E 2E 2E
6D08- 3E 27 3F 3F 37 36 36 56
6D10- 24 24 24 2D 2D 00 DB 23
6D18- 37 27 37 27 0C 25 25 25
6D20- 25 E5 27 27 27 27 27 2D
6D28- 2E 2C 2E 2E 2E 2E 2D 2D
6D30- 2D 36 36 3F 3F 3F 67 2D
6D38- 2D 25 3C 3F 3F 96 1A 37
6D40- 37 07 00 3F 24 24 24 1C
6D48- 1C 1C 1C 1C 2D 2D 15 15
6D50- 15 05 60 0C 2D 2D 1E 1E
6D58- 1E 1E 1E 36 36 36 27 3F
6D60- 24 24 E4 1C 1C 1C 2D 0E
6D68- 0E 0E 05 60 0C 2D 1E 1E
6D70- 1E 1E 36 36 06 00 01 C0
6D78- 3C 3E 3C 3E 3C 3E 3C 3E
6D80- 3C 26 28 3C 2C 3C 2C 3C
6D88- 0D 2E 2C 2E 2C 2E 2C 2E
6D90- 2E 2E 2E 2E 2E 3E 27 37
6D98- 27 04 00 3F 3F 3F 3F 24
6DA0- 24 2D 2D 2D 24 24 24 24
6DA8- AC 15 0E 0E 36 36 0E 0E
6DB0- 0E 0E 0E 3F 3F 3F 3C 3F
6DB8- 3F 3F 27 2C 2D 2D 25 24
6DC0- 24 24 AC 15 36 36 0E 0E
6DC8- 0E 0E 3F 3F 00 DB DB 03
6DD0- 20 24 24 24 24 24 24 2C
6DD8- 2D 2D 2D 2D 2D 2D F5 1E
```

```
6DE0- 1E 1E 1E 3F 3F 36 36 1E
6DE8- 1E 1E 1E 24 24 24 24 24
6DF0- 24 2D 2D 2D 2D 2D F5 1E
6DF8- 1E 3F 3F 37 36 36 1E 1E
6E00- 06 00 3F 3F 3F 24 24 24
6E08- 24 24 24 AC 15 15 15 2D
6E10- 2D 05 28 28 28 30 36 36
6E18- 36 36 36 36 3F 3F 27 3F
6E20- 3F 3F 24 24 24 24 24 15
6E28- 15 15 2D 2D 05 28 28 30
6E30- 36 36 36 36 3E 3F 07 00
6E38- 24 24 3C 3F 3F 3F 2C 2D
6E40- 2D 2D 24 24 24 35 36 36
6E48- 2E 2D 2D 2D 3E 3F 3F 3F
6E50- 36 36 36 00 DB 1B 05 28
6E58- 28 28 28 28 15 15 15 15
6E60- 15 05 E0 1C 1C 1C 1C 64
6E68- 0C 0C 0C 0C 1C 17 17 17
6E70- 17 17 E7 1C 1C 1C 1C 17
6E78- 0E 0E 0E 0E 0E F6 1E 1E
6E80- 1E 1E 06 00 03 3F C0 3E
6E88- BC 3E 3E 3C 3E 9C 2B 2C
6E90- 2C 2C 2C 2C 2C 2C 2E 2C
6E98- 2E 2C 2E 2C D6 3B 3E 3E
6EA0- 3E 00 43 3F 24 24 2D 2D
6EA8- 25 24 3C 3F 3F 24 24 2D
6EB0- 2D 2D 3E 3F 3F 36 2D 2D
6EB8- 35 36 36 3E 3F 3F 36 2D
6EC0- 2D 35 3F 3F 00 1E 2D 15
6EC8- 3F 3F 0E 2D 3E 3F 00
```

圖案圖形表

變數使用說明

ZZ …… 鍵盤的ASCⅡ碼。

S1,S2 …… 水平（S1）垂直（S2）移動狀態（第一次慢，第二次快，第三次停止）。

P1,P2 ……鍵盤輸入資料及鍵盤指標位址。

A1,B1 …… 1或 -1(作爲計算圖形移動速度的依據)。

A,B ……水平、垂直坐標的移動增量。

SF ……速度。

SX,SY ……圖形座標位置。

SW …… 1或0(作爲圖形是否開始移動的訊號)。

BN ……計算現值(螢幕)。

CT ……計算已經被重疊的數目。

SN ……內部圖形數目。

SC ……分數，最高分數。

CO ……圖形顏色。

TC ……計時。

TI ……時間圖形的X座標。

X0,Y0 ……圖形的最初位置(使用在計算外圍圖形的位置)。

CX,CY …… X0，Y0的增量。

SI …… 1或-1做爲X0,Y0 的計算依據。

I,Y……環路變數。

WO ……在副程式中，顯示圖形表之圖號。

XS,YS ……圖形的最初位置(使用計算內部圖形的位置)。

LX,LY ……字串訊息顯示位置。

SV ……圖型號碼。

RA ……亂數值。

SH ……最高分數的保持變數。

矩陣使用說明

IX(21)IY(21)…… 每個內部圖形坐標。
OX(31)OY(31)…… 每個外部圖形坐標。
SH(31)…… 包含每個造型的圖號。
P(31) …… 每個造型的坐標。
SM(31)…… 被重疊的圖形。

遊戲鍵說明

遊戲進行前：

RETURN …… 遊戲開始。
ESC …… 遊戲結束，假如是由 menu 所載入的回返 menu 程式，否則結束。

遊戲進行時：

RETURN …… 圖形的水平移動。
ESC …… 圖形的垂直移動。
RETURN 及 ESC 鍵的順序動作。

第一次按 …… 慢速移動
第二次按 …… 快速移動
第三次按 …… 停止
第四次按 …… 反向

程式說明

50—70行 ……宣布矩陣。

80—95行 ……主程式呼叫副程式。

100—150行 ……等待鍵盤輸入環路。

200—230行 ……檢查 S1 和 S2 的移動狀態是否在範圍之內，計算增量，設定新座標 SX ， SY ，消去舊的圖形，繪出新的圖形。

240—315行 ……計算副程式，並設定 SX ， SY 的增量。

400—420行 ……檢查水平座標是否超過範圍。

425—440行 ……檢查垂直座標是否超過範圍。

450—459行 ……檢查圖形的垂直位置是否相同，如果是再繼續檢查水平位置，如果都吻合的話，消去圖形，發出聲音，重訂移動狀態（ S1, S2）， CJ 值加 1 。

460—469行 ……完成和 450 ～ 459 一些相同的方程式，除了先檢查水平位置然後再垂直位置之外。

470—484行 ……被重疊的圖形。

485—489行 ……消去舊的分數，顯示新的分數。

490—499行 ……時間減少（圖形表示）。

500—575行 ……假如圖形已經開始移動（ SW = 1 ），你又按下 Space 鍵時，把圖形回歸至框外原來位置，移動指標至下一圖形。

580—595 行 ……跳過框外已無圖形的區域。

700—799 行 ……設定初值。

800—845 行 ……把資料請入矩陣。

850—899 行 ……設定內部及外圍圖形座標位置，並且存入矩陣之中。

900—965 行 ……繪出遊戲的框框，並且等待鍵盤的輸入。

970—977 行 ……設定時間。

978—984 行 ……等待鍵盤輸入。

985—989 行 ……重新設定時間。

990—992 行 ……顯示時間刻度。

1000—1070 行……選擇任何 20 個不同圖形在內圍。

1080—1099 行……在螢幕上顯示內部及外圍圖形。

1100—1145 行……顯示文字訊息副程式。

1150—1199 行……顯示分數副程式。

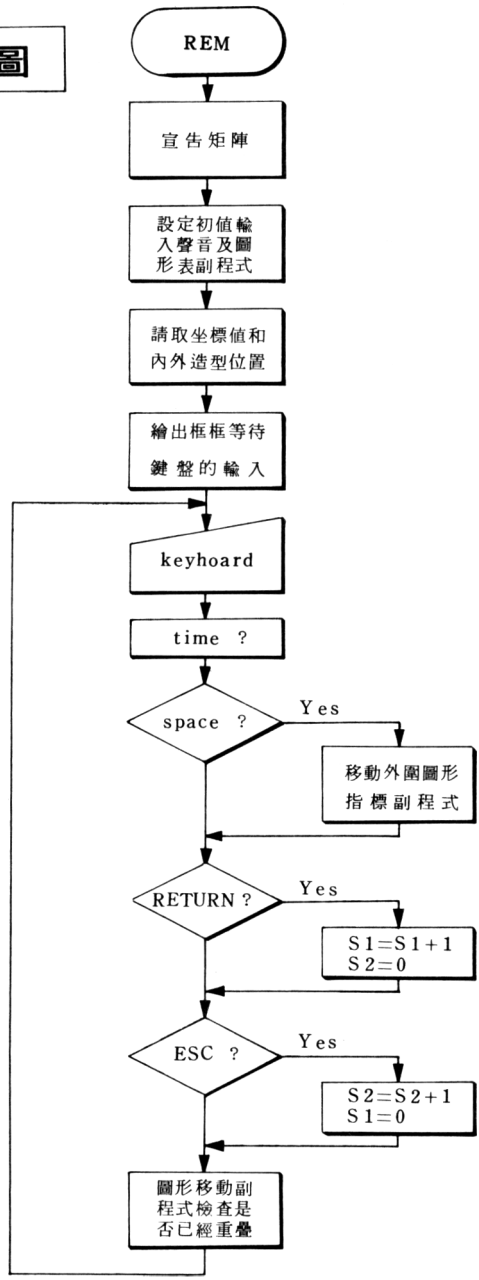
1200—1268 行……結束副程式，時間完畢時，或者是圖形全部被重疊了。假如是全部被重疊了，閃動圖形，增加分數。

1270—1286 行……等待鍵盤的鍵入是否再玩，假如是的話，重新設定變數，假如是按下 ESC ，則結束程式並且看此程式是否是 menu 所載入的，假如是的話，回返 menu 。

1290—1300 行……重新設定變數副程式。

2000—2200 行……“自動讀取圖形”的副程式。

主程式流程圖


REM
宣告矩陣
設定初值輸入聲音及圖形表副程式
請取坐標值和內外造型位置
繪出框框等待鍵盤的輸入
keyhoard
time ?
space ?
Yes
移動外圍圖形指標副程式
RETURN ?
Yes
S1=S1+1
S2=0
ESC ?
Yes
S2=S2+1
S1=0
圖形移動副程式檢查是否已經重疊

注意事項

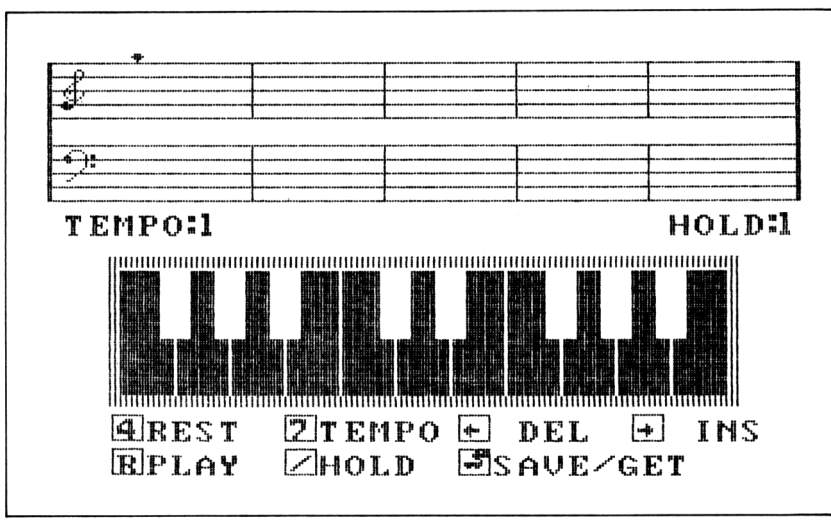
本程式如第 7 章一樣是以磁帶使用的形成寫成的，對於磁碟機的使用者，請將 31 行及 2000 副程式去掉，另外加入：

```
771 D$ =  CHR$ (13) +  CHR$ (4)
772  PRINT D$;"BLOAD MATCH-TBL"
774  PRINT D$;"BLOAD ALPHA-TBL"
```

便可以了，以後各程式的磁帶與磁片程式之變換，請依此類推。

第10章

PIANO



遊戲說明

這遊戲允許你在電腦鍵盤上彈奏及組合樂曲，使用鍵盤你可以在螢幕上的五線譜顯示你所彈奏的音符，然後你也可以按下 RETURN 鍵，聆聽你自己所譜的曲子。

你可以經由鍵盤選擇高低音符，寫入五線譜內。螢幕上的五線譜一次可顯示 20 個音符，當然你可以加上更多的音符，螢幕

上的五線譜若不敷使用，會自動的向前移動一小節空間出來，畫面上共有 4 個小節。

音符可由音符指標（箭頭）的位置、修改、消除、或從中插入，你可以按下“：”鍵使得指標移向左方，“－”使得指標移向右方。

當完成你的作品時，可以按 RETURN 鍵來聆聽自己的大作，你也可以將它儲存在磁碟中，以便以後修改它們，欣賞自己作品。注意當你從磁碟取回另一套樂曲時會喪失現存的樂曲，因此必須將剛創作的曲子存至磁碟才行。

程式

```
]LIST


10  REM  ************************
20  REM  **     PIANO  !!!     **
30  REM  ************************
32  DIM X(12),K(30)
34  DIM NY(90),NX(90)
36  DIM SV(90),S(90)
38  DIM P(160),D(160),PY(160)
40  DIM PX(160),SH(160)
48  DIM WD$(12)
50  REM  ---------------
55  REM       MAIN
60  REM  ---------------
65  GOSUB 500: GOSUB 600: GOSUB 700: GOTO
     100
```

```
100  REM  ---------------
105  REM   PIANO INPUT
110  REM  ---------------
115 ZZ =  PEEK (P2) - 128
116  IF ZZ = 33 THEN  GOTO 1250
118  IF ZZ < 0 THEN  GOTO 115
122  IF S(ZZ) = 0 THEN  GOTO 115
125  IF S(ZZ) < 9 THEN  GOSUB 180: GOTO 115
128  IF NT + 1 > NL THEN 115
130  POKE 0,S(ZZ): CALL 768
135  XDRAW SV(ZZ) AT AX,NY(ZZ)
138  IF I <  = NT THEN  XDRAW SH(I) AT AX,
     PY(I): GOTO 145
140 NT = NT + 1:NS = NS + 1
145 P(I) = S(ZZ):PY(I) = NY(ZZ):PX(I) = NX
     (ZZ):D(I) = DU * DF:SH(I) = SV(ZZ)
150  GOSUB 275
160  POKE P1,0: GOTO 115
180  REM  -- OTHER MODES --
182  ON S(ZZ) GOSUB 220,230,250,275,400,10
     00,900,900
190  POKE P1,0
199  RETURN
200  REM  --- REST ---
210  FOR T = 1 TO 100: NEXT
215  RETURN
220  REM  -- TEMPO --
221 CO = 0:SG$ =  STR$ (TF):LX = 60:LY = 8
     5: GOSUB 1200
222 TF = TF + 1: IF TF > 9 THEN TF = 1
224 CO = 3:SG$ =  STR$ (TF):LX = 60:LY = 8
     5: GOSUB 1200
225  RETURN
230  REM   -- DURATION --
231 CO = 0:SG$ =  STR$ (DF):LX = 270:LY =
     85: GOSUB 1200
```

```
232 DF = DF + 1: IF DF > 4 THEN DF = 1
234 D(I) = DU * DF
235 CO = 3:SG$ =  STR$ (DF):LX = 270:LY =
    85: GOSUB 1200
240  RETURN
250  REM  -- MOVE ARROW LEFT --
251  IF I <  = 1 THEN  GOTO 270
252  XDRAW 6 AT AX,AY
254  IF AP = 1 THEN RL =  - 1: GOSUB 800: GOTO
    260
258 AX = AX - 12:I = I - 1:AP = AP - 1
260  XDRAW 6 AT AX,AY
270  RETURN
275  REM  -- MOVE ARROW RIGHT --
276  IF NT = 0 THEN  GOTO 299
278  IF I = 1 THEN  GOTO 284
280  IF I - NT = 1 THEN  GOTO 299
284  XDRAW 6 AT AX,AY
285  IF AP = 20 THEN  GOSUB 800: GOTO 288
286 AX = AX + 12:I = I + 1:AP = AP + 1
288  XDRAW 6 AT AX,AY
299  RETURN
400  REM  ----------
405  REM   PLAYBACK
410  REM  ----------
416 N = I - 1:Y = 150
418  IF N = 0 THEN  RETURN
420  FOR J = 1 TO N
421  IF P(J) = 10 THEN  GOSUB 200: GOTO 44
    0
422  XDRAW 1 AT PX(J),Y - 4 * SH(J)
430  POKE 0,P(J): POKE 769,D(J): CALL 768
432  XDRAW 1 AT PX(J),Y - 4 * SH(J)
435  FOR T = 1 TO 10 * TF: NEXT
440  NEXT
445  POKE P1,0: POKE 769,20
```

```
449  RETURN
500  REM  ---------------
502  REM   DATA SET UP
504  REM  ---------------
505  DATA  "REST   TEMPO   DEL    INS"
506  DATA  "PLAY   HOLD   SAVE/GET"
508  DATA  "4      7      <       >"
510  DATA  "R      /      #"
511  DATA     "TEMPO:","HOLD:"
512  DATA  "ENTER TUNE TO GET ORSAVE"
513  DATA   "  (MUST START WITH P-)"
514  DATA   "SAVE:"," GET:"
515  DATA    "ENTER E TO EXIT SAVE/GET MOD
     E
516  REM   --- PIANO COORD ---
517  DATA   30,100,250,100,30,155
518  DATA  45,65,85,125,145,185,205,225
520  REM   -- ASCII KEY --
522  DATA   27,49,81,50,87,51,69,52
524  DATA  82,53,84,54,89,55,85,56
526  DATA  73,57,79,48,80,47,58,45
528  DATA  13,35,8,21
530  REM  --- SCALE VALUE ---
532  DATA  164,160,156,152,148,144
534  DATA  140,10,132,128,124,120
536  DATA  116,1,108,104,100,96
538  DATA   92,88,84,2,3,4,5,6,7,8
540  REM  -- SCALE Y POS --
541  DATA  54,54,51,51,48,48,45,42
542  DATA  42,42,39,39,36,36,33,33
543  DATA  30,30,27,27,24
545  REM  --- PIANO-X-POS ---
546  DATA  40,50,60,70,80,90,100,0,120,130
     ,140,150,160,0
547  DATA  180,190,200,210,220,230,240
548  REM  -- SHAPE NUM ----
```

```
549  DATA  1,7,1,7,1,7,1,2,1,7,1,7,1,7,1,7
     ,1,7,1,7,1
550  REM  -- READ TO ARRAYS --
552  FOR I = 1 TO 11: READ WD$(I): NEXT
554  FOR I = 1 TO 3: READ X(I),Y(I): NEXT

556  FOR I = 4 TO 11: READ X(I): NEXT
558  FOR I = 1 TO 28: READ K(I): NEXT
560  FOR I = 1 TO 28: READ K:S(K(I)) = K: NEXT

562  FOR I = 1 TO 21: READ K:NY(K(I)) = K:
      NEXT
564  FOR I = 1 TO 21: READ K:NX(K(I)) = K:
      NEXT
566  FOR I = 1 TO 21: READ K:SV(K(I)) = K:
      NEXT
570  FOR I = 1 TO 70:D(I) = 20
599  RETURN
600  REM  ----------
610  REM   INIT
615  REM  ----------
618  SPEED= 255
620 MO = 0:JX = 0:JY = 175
621 S(0) = 0:J = 1:DF = 0:TF = 0
622 DU = 25:I = 1:AX = 36:AY = 9
624 X = 30:Y = 50:RL = 1
625 NT = 0:AN = 0:AP = 1:NL = 159
630 D$ =  CHR$ (13) +  CHR$ (4)
635 P1 =  - 16368:P2 =  - 16384
642  PRINT D$;"BLOAD SOUND"
644  PRINT D$;"BLOAD PIANO-TBL"
650  PRINT D$;"BLOAD ALPHA-TBL"
660  POKE 232,120: POKE 233,105
665  POKE P1,0: POKE 769,20
670  HGR2 : SCALE= 1: HCOLOR= 3
680  REM  --TEXT PAGE --
```

```
681  HOME : PRINT "HELLO"
682  HTAB 7: VTAB 5: PRINT WD$(7)
683  HTAB 7: VTAB 6: PRINT WD$(8)
684  HTAB 15: VTAB 10: INVERSE : PRINT WD$
     (9)
685  HTAB 15: VTAB 14: INVERSE : PRINT WD$
     (10): NORMAL
687  HTAB 5: VTAB 20: PRINT WD$(11)
699  RETURN
700  REM  ------------
710  REM   SET UP
720  REM  ------------
722  REM  -- STAFF --
725 Y = 10:CO = 3
730  FOR I = 1 TO 11
731  IF I = 6 THEN  GOTO 734
732  HPLOT 4,Y TO 276,Y
734 Y = Y + 6: NEXT
735  FOR I = 1 TO 2
736  HPLOT 2 + I,10 TO 2 + I,70: HPLOT 275
      + I,10 TO 275 + I,70: NEXT
737 J = 78: FOR I = 1 TO 4
738  HPLOT J,10 TO J,34: HPLOT J,46 TO J,7
     0
739 J = J + 48: NEXT
740  DRAW 3 AT 10,30: DRAW 4 AT 10,61
741 SG$ = WD$(5):LX = 10:LY = 85: GOSUB 12
     00:SG$ = WD$(6):LX = 230: GOSUB 1200:
      GOSUB 220: GOSUB 230
742  REM     -- PIANO --
743 LN = Y(3) - Y(1):BL = X(1) + 20
744  FOR I = 1 TO LN
746  HPLOT X(1),Y(1) + I TO X(2),Y(1) + I:
      NEXT
747  HCOLOR= 0
748  FOR I = 1 TO 10
```

```
750  HPLOT BL,Y(1) TO BL,Y(3):BL = BL + 20
     : NEXT
752  FOR I = 1 TO 8
753 II = I + 3
754  FOR J = 1 TO 30
756  HPLOT X(II),Y(1) + J TO X(II) + 10,Y(
     1) + J
758  NEXT : NEXT
760  REM  -- PIANO BORDER --
762  HCOLOR= 2: FOR I = 25 TO 255
764  IF I < 30 OR I > 250 THEN  HPLOT I,95
      TO I,160: GOTO 770
768  HPLOT I,95 TO I,99: HPLOT I,156 TO I,
     160
770  NEXT
775  REM     -- BOTTOM SCREEN -
776 CO = 3:SG$ = WD$(1):LX = 40:LY = 175:
     GOSUB 1200
778 SG$ = WD$(2):LX = 40:LY = 190: GOSUB 1
     200
780 SG$ = WD$(3):LX = 29:LY = 175: GOSUB 1
     200
781 SG$ = WD$(4):LX = 29:LY = 190: GOSUB 1
     200
783  HCOLOR= 3: FOR I = 1 TO 7
784  DRAW 5 AT 27 + JX,JY:JX = JX + 63
785  IF I = 4 THEN JY = JY + 15:JX = 0
786  NEXT
788  XDRAW 6 AT 36,9
790 J = 1:I = 1
799  RETURN
800  REM  ----------------
810  REM  SET SCROLL
815  REM  ----------------
816 X = 4 - (NT - I): IF X < 0 THEN X = 0
820 NE = NS:ND = NE + 4
```

```
822  IF I = NT THEN ND = NE - 4: GOTO 830
824  IF (I < NT) AND (AP = 20) THEN ND = N
     E - X
825  IF ND > 20 THEN ND = 20
830  GOSUB 840
832  IF RL = 1 THEN I = I + 1
834 RL = 1
835  RETURN
840 IX = SC * 4 + 1:IZ = IX:XX = 36: GOSUB
     850:IX = IZ
842 AP = 17: IF RL =  - 1 THEN AP = 5
844 AX = (AP - 1) * 12 + 36
845 XX = 36:SC = SC + RL:IX = SC * 4: GOSUB
     875
849  RETURN
850  REM  -- ERASC --
855  FOR II = 1 TO NE
860  XDRAW SH(IX) AT XX,PY(IX)
870 IX = IX + 1:XX = XX + 12
872  NEXT
874  RETURN
875  REM  -- DRAW --
876  IF ND = 0 THEN  GOTO 882
877  FOR II = 1 TO ND
879  XDRAW SH(IX + II) AT XX,PY(II + IX):X
     X = XX + 12
880  NEXT
882 NS = NS - (NE - ND)
884  RETURN
900  REM  -- DEL/INS NOTE --
901  IF I > NT THEN  RETURN
904 XX = AX:SW = 9 - S(ZZ):NE = NS - AP +
     1
905  IF SW = 1 AND NT + 1 > NL THEN  RETURN
906  IF NT - I > NE THEN ND = NE
910  IF NT - I < NE THEN ND = NE - 1
```

```
912  IF SW = 1 THEN ND = NE + 1
913  REM  -----DEL/INS SOUTINES---
914  IF SW = 1 AND NS = 20 THEN ND = NE
915  ON SW GOTO 925,930
925 IX = I: GOSUB 850:IX = I: GOSUB 950:IX
      = I - 1:XX = AX: GOSUB 875: GOTO 990

930 IX = I: GOSUB 850:IX = I: GOSUB 975:IX
      = I - 1:XX = AX: GOSUB 875: GOTO 990

950  REM  -- INSERT NOTE --
952  FOR II = NT TO IX STEP  - 1
960 PY(II + 1) = PY(II):PX(II + 1) = PX(II
    )
962 D(II + 1) = D(II):SH(II + 1) = SH(II)
964 P(II + 1) = P(II)
965  NEXT
966 PY(IX) = NY(52):PX(52) = NX(52)
967 D(IX) = 20:SH(IX) = SV(52):P(IX) = S(5
    2)
970 NT = NT + 1: RETURN
975  REM  -- DELETE NOTE --
977 NT = NT - 1
978  FOR II = IX TO NT
980 PY(II) = PY(II + 1):PX(II) = PX(II + 1
    )
982 D(II) = D(II + 1):SH(II) = SH(II + 1)
984 P(II) = P(II + 1)
985  NEXT : RETURN
990  RETURN
1000  REM  -----------------
1001  REM    SAVE/GET MODE
1002  REM  -----------------
1005  POKE P1,0
1010  POKE  - 16302,0: POKE  - 16300,0: POKE
      - 16303,0
```

```
1020 HT = 20:VT = 10: POKE 216,0
1025  HTAB HT: VTAB VT: INPUT NA$
1030  IF NA$ = "E" THEN  POKE  - 16304,0: POKE
      - 16297,0: POKE  - 16299,0: RETURN
1035  IF  LEN (NA$) = 0 THEN VT = 24 - VT:
      GOTO 1025
1036  IF  MID$ (NA$,1,2) <  > "P-" THEN  GOTO
     1025
1038  ONERR  GOTO 1050
1040  IF VT = 10 THEN  GOSUB 1150: HTAB 12
     : VTAB 18: PRINT "TUNE SAVED       ": GOTO
     1020
1045  GOSUB 1100: HTAB 12: VTAB 18:PRENT"T
     UNE RETRIEVED ": GOTO 1020
1050  REM  -- ERROR --
1055  HTAB 12: VTAB 18: PRINT "ERROR
     "; PEEK (222): POKE 216,0: CALL  - 10
     621:ZZ = 35: GOTO 125
1100  REM  -- GET TUNE --
1105  PRINT D$;"VERIFY ";NA$
1110  PRINT D$;"OPEN ";NA$
1115  PRINT D$;"READ ";NA$
1120  INPUT TT:
1121  XDRAW 6 AT AX,AY:AX = 36:AY = 9: XDRAW
     6 AT AX,AY
1122 IX = SC * 4 + 1:XX = 36:NE = NS: GOSUB
     850
1123 NT = TT
1124  FOR II = 1 TO NT
1125  INPUT P(II),PY(II),PX(II),D(II),SH(I
     I)
1130  NEXT
1135  PRINT D$;"CLOSE ";NA$
1145 ND = NT: IF ND > 20 THEN ND = 20
1148 IX = 0:XX = 36:AP = 1:I = 1:SC = 0: GOSUB
     875
```

```
1149  RETURN
1150  REM  -- SAVE TUNE --
1152  PRINT D$;"OPEN ";NA$
1155  PRINT D$;"DELETE ";NA$
1160  PRINT D$;"OPEN ";NA$
1162  PRINT D$;"WRITE";NA$
1175  PRINT NT
1180  FOR II = 1 TO NT
1185  PRINT P(II);",";PY(II);",";PX(II);",
     ";D(II);",";SH(II)
1188  NEXT
1190  PRINT D$;"CLOSE ";NA$
1199  RETURN
1200  REM  --- PRINT WORDS -----
1210  HCOLOR= CO: POKE 232,168: POKE 233,9
     7
1220  FOR II = 1 TO  LEN (SG$)
1230 SV =  ASC ( MID$ (SG$,II,1))
1240  DRAW SV - 31 AT LX,LY:LX = LX + 9: NEXT

1245  POKE 232,120: POKE 233,105
1249  RETURN
1250  IF  PEEK (800) <  > 1 THEN  HGR2 : HOME
     : TEXT : HTAB 18: VTAB 10: PRINT "END
     ED": END
1260  PRINT D$;"RUN MENU"
```

請將程式 4 － 0 附加到程式上或將程式 4 － 1 以SOUND爲檔名存入磁片

圖號	1	2	3	4	5	6	7
圖案	♩	𝄾	𝄞	𝄢	□	𝅝	𝅗𝅥

圖案及其圖形號碼

```
*6978.6A11

6978- 07 00 10 00 1D 00 27 00
6980- 4C 00 68 00 7F 00 88 00
6988- 3F 28 2D 3C 3F 0C 6D 22
6990- 24 24 24 24 00 2D 2C 3C
6998- 0C E5 27 E5 3F 07 00 2D
69A0- 60 24 24 24 24 24 24 24
69A8- 0C AD 36 1E 17 9F 1E 1E
69B0- 1E 36 0E 0E 6D 05 28 20
69B8- 1C FF 17 96 1A BF 2D 35
69C0- 3F 77 05 00 2D 60 0C 0C
69C8- 0C 0C 24 24 E4 1C 3F BF
69D0- 1E 17 2D 15 3F 77 6D 49
69D8- 49 12 25 27 40 27 2D 00
69E0- 2D 2D 2D 2D 2D 25 24 24
69E8- 24 24 24 3F 3F 3F 3F 3F
69F0- 37 36 36 36 36 36 00 05
69F8- 38 3F 28 2D E5 3F 07 00
6A00- 3F 28 2D 3C 3F 0C 6D 22
6A08- 24 24 24 FC DB 9B 92 52
6A10- 07 00
```

圖案圖形表

變數使用說明

ZZ ……鍵盤的ASCⅡ值。

NT ……音符總數目。

NL ……音符旋律限制（你鍵入音符旋律的最大值）。

I ……使用在決定指標在那個音符的位置）。

NS ……在五線譜上音符的數目。

AX ……音符在五線譜上的水平位置。

AY ……指標的垂直位置。

XX ……五線譜上音符外在的水平位置（使用在畫出及消去的環路）。

CO ……色彩的選擇。

S $……顯示字串的區域。

TF ……拍子的速度，（決定你在聆聽以前所譜曲子的速度）。

DF ……持續速度。決定你所彈奏的拍子。

DU ……持續時間的長度。

LX, LY …… prirt 副程式的位置設定。

RL ……拍節的左移或者右移（右移 RL = 1 ）左移（ RL = -1 ）。

AP ……指標在五線譜的位置。

Y……在整個程式中垂直坐標變數。

N……（重覆欣賞時音符的數目）。

P1, P2 ……鍵盤位址。

LN, BL ……最初設定螢幕時　時變數。

NE ……計算因移動小節或者是修改、加入、被消除的音符數目。

ND ……計算有音符被消除後剩餘的音符數目。

X……當奇數個音符要被畫在五線譜上時的抵補計算變數。

SC ……拍子的長度。

ZX ……音符的坐標值。

矩陣使用說明

X(12) ……最初螢幕設定的座標值。

S(90) ……爲了調音用的音階數，或者爲 TEMPO 1～9 的參數。足碼的產生是 ASCII 值減 128。

NY(90)……五線譜上每個音符的垂直位置。

NX(90)……在重覆放音時在鍵盤上出現音符的水平位置。

SV(90)……所要畫出造型的號碼，足碼是 ZZ（大部份爲造型表的第一個及第七個。

P(160)……每一個音符所發出的聲音（頻率）（幾度音）。

D(160)……每一音符的拍子。

PY(160)…在五線譜上每一音符的垂直位置。

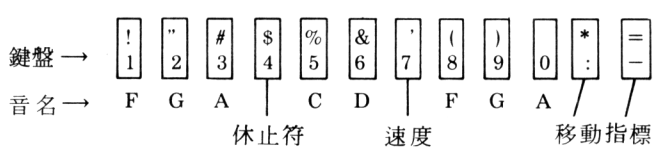
PX(160)…每一音符在重覆放音的水平位置。

SH(160)…每一音符所畫出的造型碼。

K(30) ……其值爲 ASCⅡ的變數，用以將按鍵轉變成音高及位置值。

WD$(12)…字串顯示。

遊戲鍵說明



鍵盤→	ESC	Q	W	E	R	T	Y	U	I	O	P
音名→	F	G	A	B	C	D	E	F	G	A	B

[$ 4] 休止符(Rest)……在五線譜上畫出休止符。

[, 7] 速　度(Tempo)…設定演奏的速度。

[* :] 左　移……將指標向左移動。

[= -] 右　移……將指標向右移動。

[? /] 音　長(Hold)……設定音長，音長的設定方法有下列兩種：①將指標的箭頭移到所欲設定音長的音符上，然後按[/]來改變音長的值②先按[/]鍵設定好音長，再移指標箭頭到所欲設定之音符上，然後重新鍵入該音符，便可以了。

[←] 消　去……消去目前指標箭頭所指之音符，右邊之音符會自動補齊空缺。

[→] 插　入……在目前指標箭頭所指之位置插入休止符。

[#] 存／取……指[#]後，程式進入存／取狀態，此時你可以將創作的曲子存入磁片或將磁片中的曲子取回到Apple來演奏。

[E] 畫　面……由存／取狀態回到畫面上來。

[!] 結　束……結束程式，回到節目單(假設節目單程式已在你的磁片上)。

[RETURN] ……令Apple演奏目前的樂曲。

程式說明

10— 48行 ……宣布矩陣。

50—65行 ……主要副程式。

100—160行 ……等待鍵盤的輸入，把音符畫在五線譜上，儲存音符資料在矩陣之中；檢查是否爲新插入的音符，是的話消去舊的音符。

188—199行 ……特殊功能鍵的分支點。

200—215行 ……休止符的延長時間。

220—225行 ……拍子速度（重複放音時），設定和表示 t-empo數字（1—9）。

230—240行 ……持續（拍長）設定和表示 duration 數字。

250—270行 ……移動指標往左；消除舊的指標；調整1和發出新的指標。

400—449行 ……重複放音副程式，把音符繪在 keybard 上。

500—599行 ……把資料放進矩陣內。

600—699行 ……設定初值，把聲音及圖形表副程式載入，設定爲 save 及 get 用的本又型。

700—799行 ……顯示遊戲圖形。

700—741行 ……繪出五線譜和Tempo ，Hold 文字訊息。

742—758行 ……繪出鍵盤。

760—770行 ……繪出底盤。

900—930行 ……決定在五線譜上的變數資料。消除舊的指標，繪出新的指標。

950—970行 ……插入音符副程式，把插入後所改變移動的資料存入矩陣。

975—990行 ……刪除音符副程式，把指標向左移一格，以填補被消去音符的空缺。

1000—1045行……SAVE／GET 副程式，恢復本文型，等待你再儲存，提取做一頭揖。

1050—1055行……錯誤處理副程式，顯示錯誤訊息。

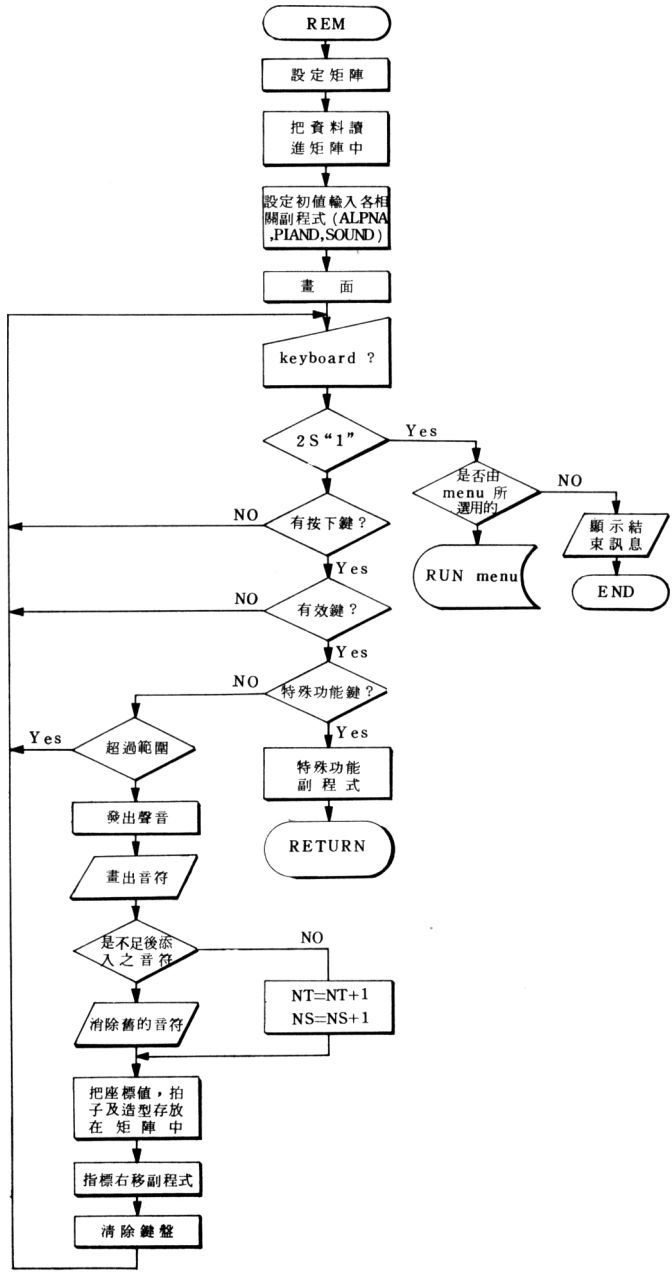
1100—1149行……GET 副程式，將檔案資料請進矩陣之中，消去現存五線譜之各音符，並繪出你所提取的樂譜。

1150—1199行……儲存副程式，將樂譜存入磁碟之中。

1200—1249行……印出副程式，將文字息顯示在高解析中。

1250—1260行……結束副程式，將程式結束，如果是由 menu 來選擇的回去 menu 。

主程式流程圖

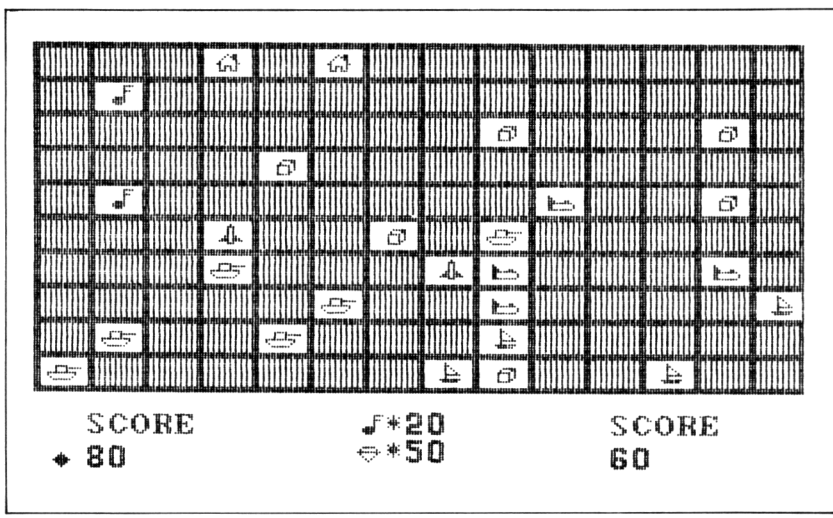

REM
設定矩陣
把資料讀進矩陣中
設定初值輸入各相關副程式（ALPNA,PIAND,SOUND）
畫 面
keyboard ?
2S"1"
Yes
是否由menu所選用的
NO
顯示結束訊息
END
RUN menu
NO
有按下鍵？
Yes
NO
有效鍵？
Yes
NO
特殊功能鍵？
Yes
Yes
超過範圍
特殊功能副程式
RETURN
發出聲音
畫出音符
是不足後添入之音符
NO
NT=NT+1
NS=NS+1
消除舊的音符
把座標值，拍子及造型存放在矩陣中
指標右移副程式
清除鍵盤

注意事項

本程式以磁片使用的方式寫成，如果是磁帶用者，除了依前幾章的方式更改之外，還要將有關樂曲存取的OPEN、READ等指令修改才行。

第11章

迷　宮



遊戲說明

迷宮是一種集中注意力和記憶力的GAME，常見於樸克牌遊戲中，遊戲一開始，在高解像畫面上有 14 × 10 的正方形方格子，這些方格之中，藏著十四種圖形，遊戲者必須去試著打開這些正方形格子，去找出另一相同的圖形。

遊戲中，你可以選擇RETURN或者是“L”(時間較長）也可

以按下ESC結束遊戲，你可以用Y、G、H、B鍵去移動指標到你所選擇的方格內，然後按下空格棒，把藏在方格內的圖形顯現出來。如果你猜對了，還是由你繼續，否則就由另外一人接手進行。

每一次你找對一組相同的圖形時，有 10 分的分數，但是有兩個例外，那就是在方格下端有兩個圖形，上面那一個是十個圖形中的任何一種，當你找出與它相同的一組圖形時有 20 分的分數，在下面的是一鑽石有 50 分的分數，在所有的方格中它只有一組而已。看誰先到達 150 分（時間較長的爲 300 分），就是勝利者。

程式

```
]LIST

10  REM  ************************
20  REM  **       PAIRS !       **
30  REM  ************************
35  DIM S(70)
40  DIM SH(141),SO(141)
42  REM  ---------------------
44  REM       POKE  SOUND
46  REM  ---------------------
48  FOR I = 0 TO 22: READ W
50  POKE 768 + I,W
52  NEXT I
54  DATA  169,10,133,1,173,48
56  DATA  192,136,208,4,198,1
57  DATA  240,8,202,208,246
58  DATA  166,0,76,4,3,96
```

```
60  REM  ----------------
70  REM        MAIN
80  REM  ----------------
82  GOSUB 2000
83  CALL  - 144
85  GOSUB 500: GOSUB 700
90  GOSUB 600: GOTO 100
100  REM  -------------
101  REM   READ KEY
102  REM  -------------
110 ZZ =  PEEK (P2) - 150
112  IF ZZ < 0 THEN  GOTO 110
114  IF S(ZZ) = 0 THEN  GOTO 110
116  GOSUB 150
118  IF EN = 1 THEN  GOTO 900
119  GOTO 110
150  REM  --- BRANCH ---
155  GOSUB 200
160  ON S(ZZ) GOSUB 250,260,270,280,325
165  GOSUB 300
170  RETURN
200  REM  -- ERASE CROSS --
205  IF OP = 1 THEN OP = 0: RETURN
210  IF SO(PS) = 1 THEN  RETURN
215  HCOLOR= 2: DRAW 13 AT X,Y
220  RETURN
240  REM  -------------
242  REM   SET UP MOVE
244  REM  -------------
250  IF Y - Y1 < 0 THEN  GOTO 299
252 PS = PS - 14
255 Y = Y - Y1: GOTO 299
260  IF X - X1 < 7 THEN  GOTO 299
262 PS = PS - 1
265 X = X - X1: GOTO 299
270  IF X + X1 > 270 THEN  GOTO 299
```

```
272 PS = PS + 1
275 X = X + X1: GOTO 299
280  IF Y + Y1 > 150 THEN  GOTO 299
282 PS = PS + 14
285 Y = Y + Y1: GOTO 299
299  POKE P1,0: RETURN
300  REM  -- DRAW CROSS--
305  IF SO(PS) = 1 THEN  RETURN
310  IF OP = 1 THEN  RETURN
315  HCOLOR= 0: DRAW 13 AT X,Y
320  RETURN
325  REM  ------------
326  REM   SHOW SHAPE
327  REM  ------------
330  IF SO(PS) = 1 THEN  RETURN
332 OP = 1
334  ON SQ GOSUB 340,350
336  RETURN
340 XZ = X:YZ = Y: GOSUB 365: GOSUB 370
342 XH = X:YH = Y:PH = PS:OP = 1
345 SQ = 2:SO(PS) = 1: RETURN
350 XZ = X:YZ = Y: GOSUB 365: GOSUB 370
352  FOR T = 1 TO 1: NEXT
354  IF SH(PH) = SH(PS) THEN  GOSUB 400: GOTO
     359
355 XZ = XH:YZ = YH: GOSUB 375
356 XZ = X:YZ = Y: GOSUB 375:SQ = 1
357  XDRAW 12 AT PX(PL) - 10,185:PL = 3 -
     PL: XDRAW 12 AT PX(PL) - 10,185
358 OP = 0:SO(PS) = 0:SO(PH) = 0
359  RETURN
360  REM  ------------
365  REM  --- OPEN ---
366  HCOLOR= 3: GOSUB 390
368  HCOLOR= 0: GOSUB 390
369  RETURN
```

```
370  REM  --- SHOW ---
371  HCOLOR= 3: DRAW SH(PS) AT X,Y
372  POKE P1,0
374  RETURN
375  REM  -----CLOSE ----
376  HCOLOR= 3: GOSUB 390
378  HCOLOR= 2: GOSUB 390
380  RETURN
390  REM  --- WINDOW ---
391  POKE 0,10: POKE 769,4: CALL 768
392 XX = XZ - 4:YY = YZ - 10
393  IF XX > 250 THEN IX = 15
394  FOR II = 1 TO 12
395  HPLOT XX,YY + II TO XX + IX,YY + II
396  NEXT :IX = 16: POKE P1,0
399  RETURN
400  REM  ----------
401  REM    MATCH
402  REM  ----------
410 SQ = 1
420 SO(PS) = 1:SO(PH) = 1
430  FOR II = 1 TO 10: POKE 0,5 * II: POKE
     769,20: CALL 768: NEXT
440 OS = SR(PL)
450  IF SH(PS) = RS THEN SR(PL) = SR(PL) +
     10: CALL 768: CALL 768
460  IF SH(PS) = 11 THEN SR(PL) = SR(PL) +
     40: CALL 768: CALL 768: CALL 768
470 SR(PL) = SR(PL) + IC:CS = SR(PL):LX =
     PX(PL):LY = 185: GOSUB 850
475  REM  -- CHECK SCORE --
476  IF SR(1) > HS OR SR(2) > HS THEN EN = 1
499  RETURN
500  REM  ----------
501  REM   INITIALIZE
502  REM  ----------
```

```
505 X = 7:Y = 147: SPEED= 255
510 PX(1) = 20:PX(2) = 210
515 X1 = 20:Y1 = 15:IX = 16
520 SX = 0:SY = 0:HS = 149
525 SN = 10:SQ = 1:IC = 10
530 SE = 1:PS = 127:OP = 0
535 FL(0) = 1:PL = 1
545 P1 =  - 16368:P2 =  - 16384
550  DEF  FN RD(UL) =  INT ( RND (1) * UL)

555  IF EN = 1 THEN  GOTO 590
575  IF  PEEK (800) <  > 1 THEN R =  RND (
      -  PEEK (78) -  PEEK (79) * 256)
580  HGR2 : SCALE= 1: HCOLOR= 3
585  POKE 232,120: POKE 233,105
590  POKE P1,0: ROT= 0
599  RETURN
600  REM  ----------
601  REM   SET UP GRID
602  REM  ----------------
603  IF EN = 1 THEN EN = 0: GOTO 626
604 J = 0
605  HCOLOR= 2: HPLOT 0,0: CALL 62454: HCOLOR=
     0
606  FOR I = 1 TO 40
607  HPLOT 0,151 + I TO 279,151 + I: NEXT

608  HCOLOR= 3: FOR I = 0 TO 1
609  HPLOT 0 + I,0 TO 278 + I,0 TO 278 + I
     ,152 TO 0 + I,152 TO 0 + I,0: NEXT
610  REM  --- VERT LINE ---
612  FOR I = 0 TO 1
614  HPLOT SX + I,0 TO SX + I,152: NEXT
615  GOSUB 620
616 SY = SY + 15:SX = SX + 20
618  IF SX > 270 THEN  GOTO 626
```

```
619  GOTO 610
620  REM  --- HORZ LINE ---
621  IF SY > 150 THEN  RETURN
622  FOR I = 0 TO 2
624  HPLOT 0,SY + I TO 279,SY + I: NEXT
625  RETURN
626  REM  -- SET SHAPES(GRID) --
630  FOR I = 1 TO 140:J = J + 1
635 SH(I) = J: IF J = 10 THEN J = 0
640  NEXT
645  GOSUB 780
650  GOSUB 680
655  FOR I = 140 TO 2 STEP  - 1
660 J =  FN RD(I) + 1:T = SH(I)
665 SH(I) = SH(J):SH(J) = T
670  NEXT : GOTO 695
680  REM  -- PLACE DIAMON --
682 RS =  FN RD(10) + 1
684 R1 =  FN RD(13) + 1
686 R2 =  FN RD(13) + 1: IF R2 = R1 THEN  GOTO
     686
688 SH(R1 * 10 + RS) = 11:SH(R2 * 10 + RS)
      = 11
690  RETURN
695  HCOLOR= 3: DRAW RS AT 118,170: DRAW 1
     1 AT 118,182:CR = 3:WD = 4:LX = 128:L
     Y = 170: GOSUB 800
696 WD = 5:LY = 182:LX = 128: GOSUB 800
697 WD = 1:LX = 20:LY = 170: GOSUB 800:LX =
     210: GOSUB 800
698  HCOLOR= 0: DRAW 13 AT X,Y: XDRAW 12 AT
     10,185
699  RETURN
700  REM  --------
705  REM    DATA
710  REM  --------
```

```
715  REM  --- KEYS ---
720  DATA  67,49,50,44,10
725  REM  -- WORD STRINGS --
730  DATA  "SCORE","RETURN/PLAY","ESC/END"
     ,"*20","*50"
735  REM  - READ ARRAY DATA -
740  FOR I = 1 TO 5: READ K:S(K) = I: NEXT

745  FOR I = 1 TO 5: READ WD$(I): NEXT
779  RETURN
780  REM  -- BEGIN --
781 CR = 3:WD = 2:LX = 90:LY = 170: GOSUB
     800
782 WD = 3:LX = 90:LY = 185: GOSUB 800: POKE
     P1,0
783 ZZ =  PEEK (P2)
785  IF ZZ = 141 THEN HS = 149: GOTO 795
786  IF ZZ = 155 THEN  GOTO 1000
788  IF ZZ = 204 THEN HS = 299: GOTO 795
790  GOTO 783
795  IF EN = 1 THEN  CLEAR : GOTO 10
796 CR = 0:WD = 2:LX = 90:LY = 170: GOSUB
     800
798 WD = 3:LX = 90:LY = 185: GOSUB 800
799  RETURN
800  REM  ------------
801  REM  PRINT WORDS
802  REM  ------------
805  POKE 232,168: POKE 233,97
810  HCOLOR= CR
815  FOR II = 1 TO  LEN (WD$(WD))
820 SV =  ASC ( MID$ (WD$(WD),II,1))
825  DRAW SV - 31 AT LX,LY
830 LX = LX + 8: NEXT
835  POKE 232,120: POKE 233,105
845  RETURN
```

```
850  REM  -- PRINT SCORE --
851  POKE 232,168: POKE 233,97
852 SC = OS: HCOLOR= 0: GOSUB 855
853 LX = PX(PL)
854 SC = CS: HCOLOR= 3: GOSUB 855: GOTO 89
    0
855 SC$ =  STR$ (SC)
860  FOR II = 1 TO  LEN (SC$)
865 SV =  ASC ( MID$ (SC$,II,1))
870  DRAW SV - 31 AT LX,LY
875 LX = LX + 8: NEXT
880  RETURN
890  POKE 232,120: POKE 233,105
899  RETURN
900  REM  ----- END/CLEAR ---
910  FOR II = 1 TO 20: CALL 768: XDRAW 12 AT
    PX(PL) - 10,185
920  FOR T = 1 TO 30: NEXT : NEXT
930  FOR JJ = 0 TO 9: FOR MM = 0 TO 13
940 KK = KK + 1: IF SO(KK) = 1 THEN XZ = 2
    0 * MM + 7:YZ = 15 * JJ + 12: CALL 76
    8: GOSUB 375
950  NEXT : NEXT
960  FOR II = 1 TO 30
970  HCOLOR= 0: HPLOT 0,155 + II TO 279,15
    5 + II: NEXT
980  CLEAR :EN = 1: GOTO 10
1000  REM  ----------
1010  REM   ENDING
1020  REM  ----------
1030  IF  PEEK (800) <  > 1 THEN  HGR2 : HOME
    : TEXT : HTAB 18: VTAB 10: PRINT "END
    ED": END
1999  END
2000  REM   -----AUTO READ BEGIN !!!----
2010  DIM Y$(100)
```

```
2011  HOME : INVERSE : VTAB 10: HTAB 11
2015  FOR D = 0 TO 500: NEXT D: PRINT "WAI
     T !!! PLEASE !!!": NORMAL
2020 Y$ = "61A8.6FFFR N D823G"
2030  FOR I = 1 TO  LEN (Y$)
2040  POKE 511 + I, ASC ( MID$ (Y$,I,1)) +
     128
2050  NEXT I
2060  POKE 72,0
2070  RETURN
```

圖號	1	2	3	4	5	6	7
圖案							

圖號	8	9	10	11	12
圖案					

圖案及其圖形號碼

```
6978- 0D 00 1C 00 31 00 41 00
6980- 53 00 71 00 85 00 96 00
6988- AA 00 BD 00 D3 00 EA 00
6990- FF 00 10 01 09 2D 2D 2D
6998- 38 3F 3F 3F 4C 2C 2D 2D
69A0- 1C 1C 1C 1C 1C 36 36 36
69A8- 00 49 2D 65 0C C4 20 37
```

```
69B0- DF 27 6D 92 BF 1A 1C 04
69B8- 00 49 2D 05 38 3F 27 2D
69C0- 2D 3C 3F 4C 21 24 2C 2D
69C8- D6 3F 00 49 2D E5 27 25
69D0- 2D 25 DF 3F 28 2D 6C 24
69D8- 17 27 3F FF 2E 35 2E 2D
69E0- DD DB 03 B0 4A 39 3F 36
69E8- 00 21 3C 28 64 0C 0C 15
69F0- 0E 24 35 36 15 37 36 3F
69F8- 24 3F 36 3F 00 49 E1 1C
6A00- 1C 1C 64 0C AD 05 28 15
6A08- 15 F6 1E 1E 06 00 2D 2D
6A10- 2D 2D 05 20 1C 27 37 3F
6A18- 3F 3C E4 36 2E 3E 2E 1E
6A20- 06 00 01 28 2D 3C 67 25
6A28- 24 64 2E 36 36 AD 3F 2E
6A30- 2D DE 3B 3F 00 09 24 24
6A38- 0C 0C 2D 2D 36 36 1E 1E
6A40- 3F 27 08 18 28 2D 0C D6
6A48- 36 06 00 2D 2D 2D 0C 0C
6A50- 3C 3F 3F 3F 3F 37 0E 44
6A58- 08 24 2D 2D 36 0C 2D 2D
6A60- 05 00 49 E1 1C 1C 07 60
6A68- 0C 2D 2D 0E 15 3F 3F 3F
6A70- 77 49 09 1E 1E 06 00 05
6A78- 38 E7 2D 2D 0C 3F 3F 3F
6A80- 0C 2D 2D 1C 3F 0C 05 00
6A88- 49 2D 3C FF 28 2D 2D 25
6A90- 3F 3F 3F 2C 2D 2D E5 3B
6A98- 27 2D 05 00
```

圖案圖形表

變數使用說明

ZZ ……ASCII 按鍵值。

X,Y ……X , Y座標的增量。

X1 , Y1 ……文字的座標位置。

SX, SY ……設定方格的開始位置。

XH, YH ……暫存區。

HS ……最高分數限制（先到達 150? or 300?）。

P1, P2 ……鍵盤輸入資料及鍵盤選通位址。

SQ ……做爲打開的次數（第一次或第二次）的旗號。

OP ……爲 1 時表示方格已經打開（0 表示關上）。

PS ……現在指標在方格中的位置（# 1—140）。

DH ……暫存區。

SN ……圖形總數。

PL ……玩的人數（1 或 2）。

SC ……分數（使用在印出副程式）。

OS ……被消除的舊分數。

CS ……現在分數。

LX ……格子的寬度。

IC ……分數增加值。

SV ……計算展現的圖號。

CR ……顏色。

WD ……文數字字串。

EN ……結束旗號。

RS …… 20 分的圖形號碼。

矩陣使用說明

SH(141) ……每個方格內的圖號。

SR(2) ……玩者的分數。

PX(2) ……玩者分數的水平座標。

S(70) ……從DATA取得的ASCII 值，例如S(67)＝1 ,S(49)＝2。

WD$(10)……文字串及各種訊息。

SO(141)……當被猜對時，該方格所對應的SO 值爲1。

遊戲鍵說明

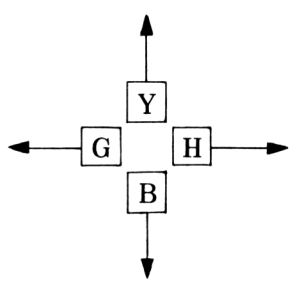
RETURN ……遊戲開始。

L ……設定長時間遊戲（最高分＝ 300 分）。

ESC ……結束。

空格鍵 ……打開方格窗戶。

移動方向鍵：



程式說明

10—50行……矩陣宣告。

60—90行……副程式進入點。

100—119行……輸入ASCII碼，並檢查是否結束了（分數超過）。

150—170行……消去舊的螢幕指標，計算出新的指標位置並且畫出來。

200—220行……消去舊螢幕指標，假如格子已經被打開，則不用，因爲格子的圖形已展現出來。

240—299行……計算新螢幕指標位置。

390—320行……畫出的螢幕指標。

325—359行……判斷格子是否被打開，假如不經打開，就把格子後的圖形顯現出來。

360—399行……把格子被打開或關起的情況顯現出來。

400—499行……假如猜對了，發出聲音，增加分數，並且檢查是否已經到達標準分數（150 or 300）。

500—599行……設定初值。

600—625行……在高解像畫面畫出格子。

626—670行……諭定各方格內的基本圖形，並且等待你選擇L、RETURN 或 ESC ，然後將格子內的圖形用亂數使之作不規則排列。

680—690行……設定鑽石在方格內的位置。

695—699行……把分數及特殊分數（20 or 50）圖形畫出來。

700—779行……把各種資料請進矩陣之中。

780—799行 ……等待鍵盤的輸入。

800—845行 ……顯示文字訊息。

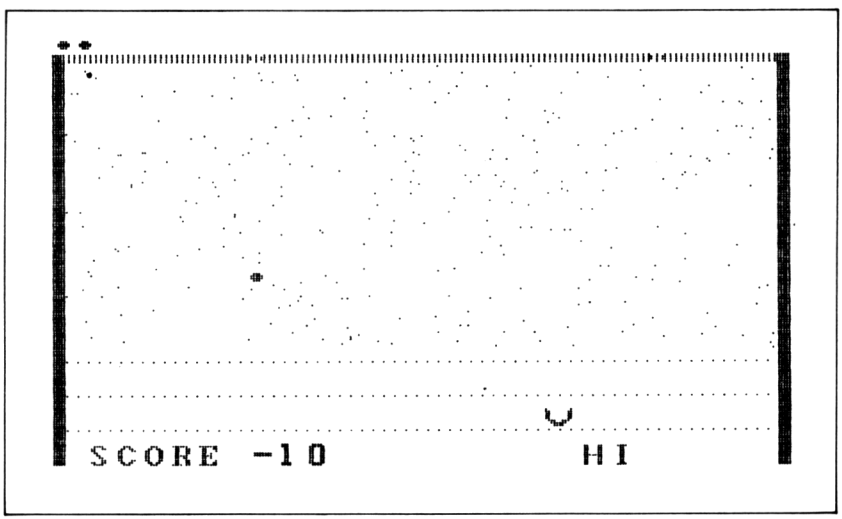
900—980行 ……關上已經被打開的所有窗戶，重新的設定結束旗號，清除矩陣資料。

注意事項

本程式是爲磁帶使用者設計的，磁片使用者請依前述方法修改之。在存錄圖形表於磁帶時，請注意所有程式一律用61A8・6FFFW，這是因爲主程式2020行用的是61A8・6FFFW而不是眞正的圖形資料長度。

第12章

摘 星 星



遊戲說明

這個遊戲，是要你使用搖桿控制畫面底端的捕手，捕捉那些由天而降的星星，將搖桿順時鐘或逆時鐘的移動，控制著捕手的左右移動。 捕手可以選擇在三個不同高度的水平線上移動，當你按下 PDL(0)時捕手將會在星星墜落時由上而下的移動在各個水平線上。當然，捕手所在的位置會影響你分數高低，在較高的

垂直位置時分數較高，較低的垂直位置分數較低，要是你沒有接到隕石，分數是相對要扣分的。

在遊戲的一開始，當按下遊戲按鈕（ 0 ）時會有一段暫停時間然後再按下遊戲按鈕（ 0 ），這遊戲就開始了。你共可以擁有三個捕手，且你的捕手每 250 分增加一個，遊戲的最高分是 6000 分。當捕手抓住星星時，你必須按下遊戲按鈕（ 0 ），才能接住星星，否則便會漏接而失誤了。

程式

```
]LIST

10  REM  **********************
20  REM  **      CATCH!!!      **
30  REM  **********************
35  DIM X1(15)
40  REM =====POKE SOUND=====
41  FOR I = 0 TO 22: READ WW
42  POKE 768 + I,WW: NEXT I
43  DATA  169,10,133,1,173,48
44  DATA  192,136,208,4,198,1
45  DATA  240,8,202,208,246
46  DATA  166,0,76,4,3,96
50  REM
60  REM  ----------------
70  REM          MAIN
80  REM  ----------------
90  GOSUB 500: GOSUB 600
95  GOSUB 700: GOTO 100
100  REM  ----------------
```

```
101  REM   READ INKEY
102  REM  ----------------
105 I = 1:BX = 26:NU = 1
110 Q =  PDL (0):IC =  INT ( ABS (W - Q) /
    WW) *  SGN (Q - W)
120  GOSUB 200: GOSUB 225: GOSUB 250: GOSUB
    300: GOSUB 400
130  GOTO 110
200  REM  -- DRAW CATCHER --
205  IF X + IC > 270 OR X + IC < 10 THEN
     RETURN
210  XDRAW 2 AT X,Y(J):X = X + IC: XDRAW
    2 AT X,Y(J)
215  RETURN
225  REM  -- LEVEL --
230  IF SW = 0 THEN  RETURN
235  IF  PEEK (PB) < 128 THEN  RETURN
240  IF J = 1 OR J = 3 THEN A = A *  - 1
245  XDRAW 2 AT X,Y(J)
249 J = J + A: XDRAW 2 AT X,Y(J): RETURN

250  IF SW = 0 THEN  GOTO 299
255  XDRAW 1 AT X1(I),Y1:Y1 = Y1 + FT: XDRAW
    1 AT X1(I),Y1
299  RETURN
300  REM  -- STAR CAUGHT ? --
305 TI = TI - 1
310  IF Y1 > YE THEN SW = 0:TI = 0: GOSUB
    450:SW = 1
315  IF SW = 0 THEN  RETURN
320  IF  ABS (YL(J) - Y1) < VR AND  ABS (
    X - X1(I)) < HR THEN  XDRAW 1 AT X1(
    I),Y1:Y1 = Y(J) - VA:X1(I) = X: XDRAW
    1 AT X1(I),Y1:SW = 0:TI = LP:XH = X1
    (I):YH = Y1
325  RETURN
```

```
400  REM  --- NUTRALIZE --
405  IF SW = 1 THEN  RETURN
410  IF TI < 0 THEN SW = 1
415  IF  PEEK (PB) > PV THEN  GOSUB 420: GOSUB
     450: RETURN
419  GOTO 449
420  REM  --- SOUND --
422  POKE 0,5: POKE 769,10: XDRAW 1 AT XH
     ,YH
424  FOR T = 1 TO 10: CALL 768: XDRAW 2 AT
     XH,YH: NEXT
425 SW = 1: RETURN
449  RETURN
450  REM  -- ADD SCORE --
452 NX = 80:CO = 0: GOSUB 475
454  IF SW = 1 THEN SC = SC + PT(J)
456  IF SW = 0 THEN SC = SC - PT(J): HCOLOR=
     0: DRAW 1 AT BX,BY:BX = BX - 8
460 CO = 3:NX = 80: GOSUB 475
462  IF SC > EB * 250 THEN EB = EB + 1:BX
      = BX + 8: GOSUB 715:BX = BX - 8
464  IF BX = 2 THEN  XDRAW 1 AT X1(I),Y1:
      GOTO 800
466  IF SW = 0 THEN  XDRAW 1 AT X1(I),Y1
468  IF I > 10 THEN  GOSUB 490:I = 0
474 Y1 = 13:I = I + 1: RETURN
475  REM  -- SCORE PRINT --
477  POKE 232,168: POKE 233,97
480 SC$ =  STR$ (SC): HCOLOR= CO
482  FOR K = 1 TO  LEN (SC$)
483 SV =  ASC ( MID$ (SC$,K,1))
484  DRAW SV - 31 AT NX,NY:NX = NX + 10: NEXT
485  POKE 232,120: POKE 233,105
486  RETURN
490 R = 1:CO = 3:WD = 5:LX = 130:LY = 190
     : GOSUB 850
```

```
492 BX = BX + 8: GOSUB 715:BX = BX - 8: GOSUB
    720
494 CO = 0:WD = 5:LX = 130: GOSUB 850
496  IF FT > 9 THEN FT = 10
498 FT = FT + 1: RETURN
500  REM  ----------------
501  REM      INITIALIZE
502  REM  ----------------
505 A = 1: SPEED= 255
510  ROT= 0:HR = 6:VR = 10:VA = 2:LP = 10

515 RL = 260:BX = 10:BY = 7:NU = 3:EB = 1

520 W = 125:WW = 12:TI = 0:PV = 127
525 X = 17:Y = 142:NY = 190
530 X1 = 100:Y1 = 13:YE = 179
535 P1 =  - 16368:P2 =  - 16384:PB =  - 1
    6287
540 FT = 6:SW = 1:YL = 140
545  DEF  FN RD(RL) =  INT ( RND (1) * RL
    )
550 D$ =  CHR$ (13) +  CHR$ (4)
555  PRINT D$;"BLOAD CATCH-TBL"
560  PRINT D$;"BLOAD ALPHA-TBL"
570  POKE 232,120: POKE 233,105
575  POKE P1,0
580  POKE 0,15: POKE 769,20
585  HGR2 : SCALE= 1: HCOLOR= 3
590  IF  PEEK (800) <  > 1 THEN R =  RND
    ( -  PEEK (78) -  PEEK (79) * 256)
599  RETURN
600  REM  -- DATA --
605  DATA  50,25,10
610  DATA  5,10,5,190,5,1
615  DATA  270,10,270,190,5,1
620  DATA  9,10,9,13,261,2
```

```
625  DATA   10,145,10,145,259,4
630  DATA  10,160,10,160,259,4
635  DATA  10,175,10,175,259,4
640  DATA  143,158,173
645  DATA  132,150,164
650  REM  --     WORDS     --
655  DATA  "PDL/START","ESC/END","SCORE",
     "HI"
660  DATA  "*BONUS*"
665  REM  -- READ ARRAYS --
670  FOR I = 1 TO 3: READ PT(I): NEXT
675  FOR I = 1 TO 6
680  READ XB(I),YB(I),XE(I),YE(I),LG(I),S
     P(I): NEXT
685  FOR I = 1 TO 3: READ Y(I): NEXT
690  FOR I = 1 TO 3: READ YL(I): NEXT
695  FOR I = 1 TO 5: READ WD$(I): NEXT
699  RETURN
700  REM  -- SETUP --
701  FOR J = 1 TO 6: GOSUB 750: NEXT
702 CO = 3
703  GOSUB 730: GOSUB 735: GOSUB 745
704 J = 3: DRAW 2 AT X,Y(J)
705  GOSUB 720: GOSUB 725: GOSUB 715: GOTO
     799
715  HCOLOR= 3: FOR II = 1 TO NU: POKE 0,
     15: POKE 769,2
716  FOR T = 1 TO 20: NEXT
717  IF BX > 250 THEN II = 3 GOTO 719
718  CALL 768: DRAW 1 AT BX,BY:BX = BX +
     8
719  NEXT : RETURN
720  REM  -- GENERATE 10 STARS --
722  FOR I = 1 TO 11:X1(I) =  FN RD(255) +
     15: NEXT
724  RETURN
```

```
725  FOR I = 1 TO 300:SX =  FN RD(RL) + 1
     0:SY =  FN RD(125) + 15: HPLOT SX,SY
726  POKE 769,1: CALL 768: NEXT
729  RETURN
730  REM   -PRINT START MESSAGE -
731 WD = 1:LX = 25:LY = 190: GOSUB 850
732 WD = 2:LX = 180: GOSUB 850
734  RETURN
735  REM  -- START --
736  IF  PEEK (P2) = 155 THEN  GOTO 1000
737  IF  PEEK (PB) < 128 THEN  GOTO 735
738  FOR T = 1 TO 20: NEXT
739  IF  PEEK (PB) < 128 THEN  RETURN
740  GOTO 735
745  REM  - PRINT "SCORE/HI" -
746 CO = 0: GOSUB 730:CO = 3:WD = 3:LX =
     20:LY = 190: GOSUB 850
748 WD = 4:LX = 200: GOSUB 850
749  RETURN
750  REM  -----DRAW PLAYING SCEEN ---
755  FOR I = 1 TO LG(J) STEP SP(J)
760  HPLOT XB(J) + I,YB(J) TO XE(J) + I,Y
     E(J): NEXT
780  RETURN
799  RETURN
800  REM  -- END --
810  POKE 0,200: POKE 769,200: CALL 768: CALL
     768
820  IF SC > PS THEN SH = SC:SC = PS:NX =
     230:CO = 0: GOSUB 475:CO = 3:SC = SH
     :NX = 230: GOSUB 475:PS = SC
830  CALL  - 10621:NU = 3:BX = 10
840 Y1 = 13:FT = 6:SW = 1:EB = 1
845  GOSUB 735: GOSUB 720: GOSUB 715:CO =
     0:NX = 80: GOSUB 475:SC = 0: GOTO 10
     0
```

```
850  REM  --- PRINT WORDS ---
860  POKE 232,168: POKE 233,97
865  HCOLOR= CO
870  FOR II = 1 TO  LEN (WD$(WD))
875 SV =  ASC ( MID$ (WD$(WD),II,1))
880  DRAW SV - 31 AT LX,LY
885 LX = LX + 10: NEXT
890  POKE 232,120: POKE 233,105
895  RETURN
1000  REM  -- END/LINK --
1010  IF  PEEK (800) <  > 1 THEN  HGR2 : HOME
     : TEXT : HTAB 18: VTAB 10: PRINT "EN
     DED": END
1020  PRINT D$;"RUN MENU"
1100  REM  ================
```

圖號	1	2
圖案		

圖案及其圖形號碼

```
*6978.699E

6978- 02 00 06 00 11 00 FD 07
6980- 28 2D 25 3F 3F 0C 2D 05
6988- 00 25 2D 08 20 24 BC 36
6990- 36 3F DF 3F 24 24 1C 36
6998- 36 56 29 2D 1E 3F 00
```

圖案圖形表

變數使用說明

Q……搖桿值（0～255）。
IC ……捕手的水平移動增量。
W……搖桿中間值（125），停住。
WW ……速度：決定IC的增加值。
X……捕手的水平座標位置。
SW ……SW(1):星星隕落。
　　(0):抓住星星。
PB ……－16287，遊戲按鈕（0）的位置。
PV ……檢查搖桿數值是否超過 127 。
J……捕手的水平位階（1,2, 或 3）。
YI ……隕石的垂直座標。
FT ……隕石墜落的垂直增量速度。
TI ……隕石停留在捕手上而未按按鈕（0）的計時。
LP ……JI 的環路計數初值。
VR,HR ……檢查是否星星被抓住。
XH,YH ……暫存變數。
CO ……顏色。
NX,NY ……分數的座標位置。
BX,BY ……隕石的座標位置。
SC ……分數。
SC$ ……分數字串。
SU ……圖形號碼。

WD ……文字變數號碼。

I ……決定下一次隕石墜落位置。

矩陣使用說明

XI(15)……由亂數所生的隕石位置。

Y(3)……捕手的水平位階之垂直座標。

XB(6),YB(6) ……圖面範圍座標。

XE(6),YE(6) ……圖面範圍座標。

YL(3) ……每個垂直水平分量的垂直座標限制，幫助了解星星是否已被捕捉。

WD$(5)……文字字串。

遊戲鍵說明

遊戲前：

遊戲按鈕(0)……遊戲開始。

ESC ……遊戲結束，假如是由 menu 提取到回返 menu 程式。

遊戲時：

搖桿(0)……控制捕手的水平移動。

遊戲按鈕(0)……改變捕手的高度，如果捕手已捕捉到星星，則按下按鈕(0)才能將星星銷毀。

程式說明

10—50行 ……宣布矩陣。
60—95行 ……主程式呼叫副程式。
100—130行 ……請取 paddle 值，並且計算 IC 的增量。
200—215行 ……檢查移動範圍，消去舊位置的捕手，顯示新位置的捕手。
225—299行 ……按鈕（0）被按下了嗎？假如是的話，移動捕手到新的垂直水平。
300—325行 ……隕石沒有被捕捉嗎？設定分數，設定 SW 和 JI。
400—449行 ……隕石在捕手裏面嗎？按鈕被按下了嗎？假如是的話，消去隕石，增加分數。
450—474行 ……分數副程式。
475—486行 ……印出分數。
490—498行 ……由亂數產生額外的（獎分）隕石位置。
500—599行 ……設定初值，輸入圖形表副程式。
600—699行 ……請取資料。
700—799行 ……設定顯示螢幕。
715—719行 ……畫出增加的捕手。
720—724行 ……由亂數產生隕石位置。
725—729行 ……背景星星副程式。
730—734行 ……顯示開始訊息。
735—740行 ……等待按鍵輸入。

745—749行 ……消去開始訊息，印出SCORE 及HISCORE 。

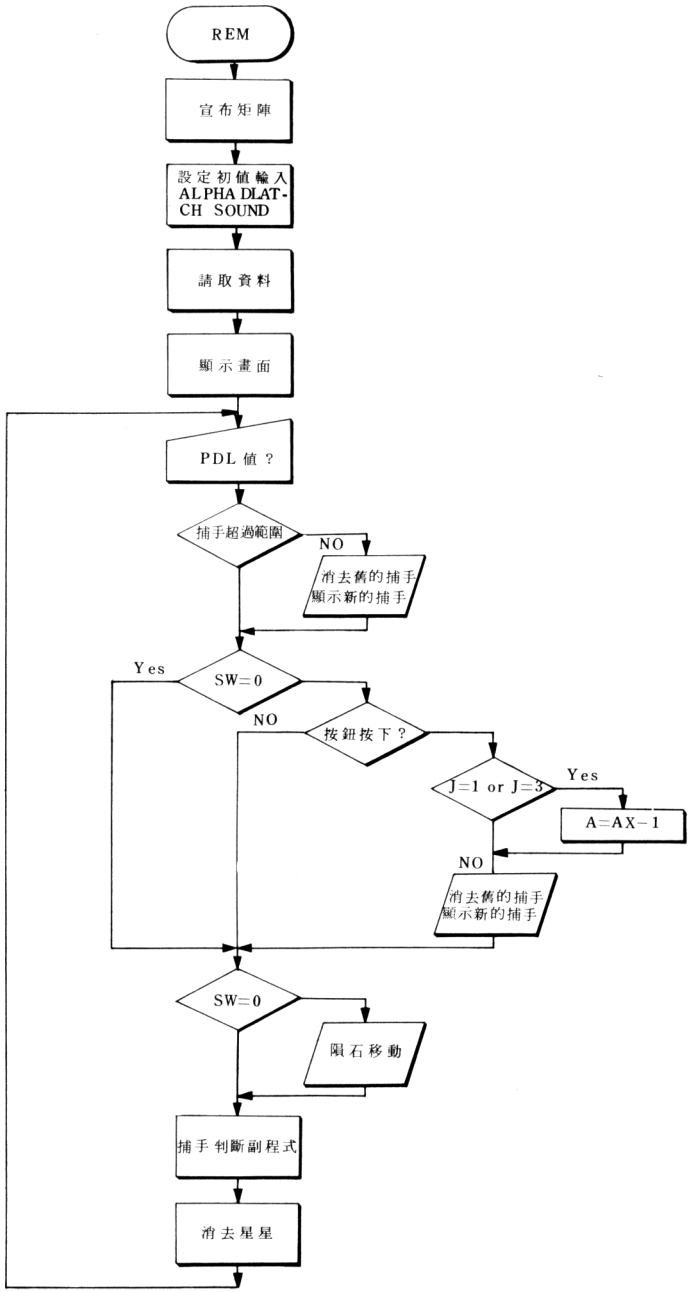
750—799行 ……顯示圖形。

800—845行 ……結束副程式，檢查是否爲最高分，清除堆疊（CALL—10621），要再玩則重新設定變數。

850—895行 ……印出文字訊息副程式。

1000—1100行……程式結束，假如是由 menu輸 入提取的話 RUN menu。

主程式流程圖

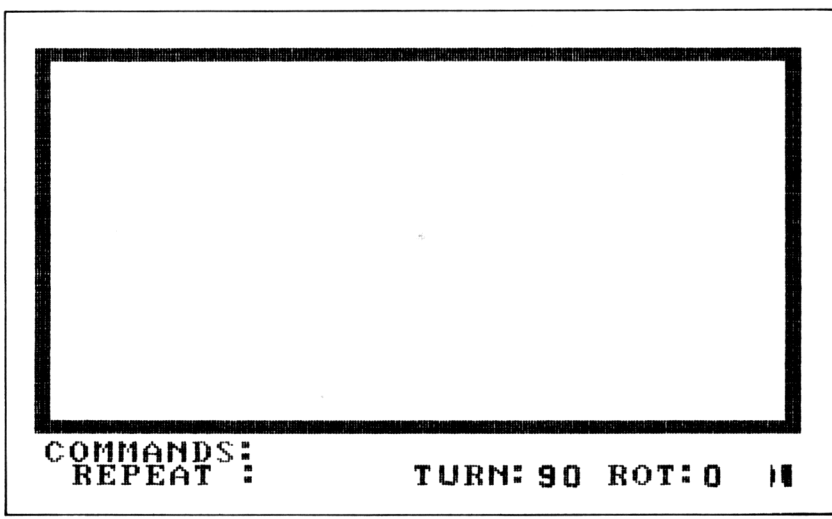

REM
宣布矩陣
設定初值輸入
ALPHA DLAT-
CH SOUND
請取資料
顯示畫面
PDL 值？
捕手超過範圍
NO
消去舊的捕手
顯示新的捕手
Yes
SW=0
NO
按鈕按下？
J=1 or J=3
Yes
A=AX-1
NO
消去舊的捕手
顯示新的捕手
SW=0
隕石移動
捕手判斷副程式
消去星星

注意事項

本程式爲磁碟機用者所設計，磁帶使用者請依前述方法更改之。

第13章

螺　　旋



遊戲說明

螺旋是一個在高解像畫面畫出螺旋狀圖形的繪圖遊戲。使用R（右）及L（左）指令，可以來控制畫圖的方向，當設定好畫圖方向（左，右）後按下 RETURN ，游標便移到底下一行，在這一行你可以下達重覆次數（ REPEAT)的指令，再按下 RETURN ，便可以在畫面上看到依照你的命令所畫出來的圖形

了。

除了左、右轉之外，你還可以做45°～90°的轉動，並且也可以做整個圖形45°的旋轉。至於圖形的大小也可以用1～5的數值來控制。使用Y、G、H、及B鍵，可以將畫圖原點移動，這樣可以在畫面上畫出多組圖形來。下達轉彎命令時，使用空格便可以令線條沿目前方向進行而不轉彎。

設計例

下圖便是一些圖形實例，指令表中的所列的便是造出這些圖形的指令：

	命令	重覆次數	轉角	圖形轉動	放大
圖 1	RRRRLRRRR	8	45	0	2
	RRRRLRRRR	8	45	0	1
	RRRRLRRRR	4	90	0	1
圖 2	RRRRRRRRRLL	4	90	0	1
	RRRRRRRRRLL	4	90	0	3
	RRRRRRRRRLL	4	90	45	1
圖 3	RRRLRRR	4	90	0	1
	RRRLLLRRR	8	45	0	1
	RRRLRRR	8	45	0	3
	RRRRLRRRR	8	45	0	1

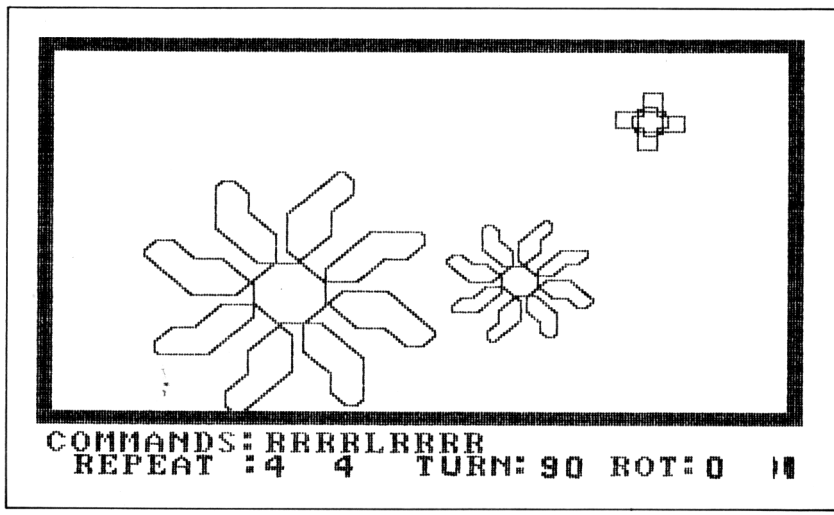

COMMANDS: RRRRLRRRR
REPEAT :4 4 TURN: 90 ROT: 0

圖 2

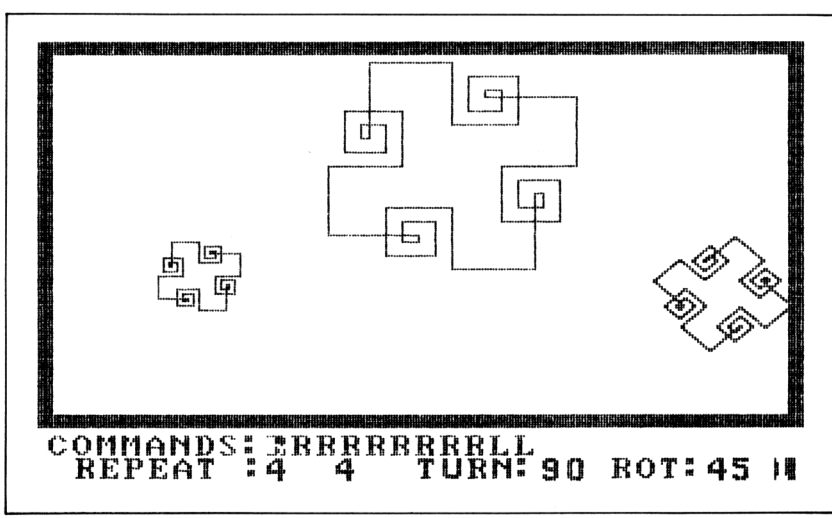

COMMANDS: RRRRRRRRRLL
REPEAT :4 4 TURN: 90 ROT: 45

圖 1

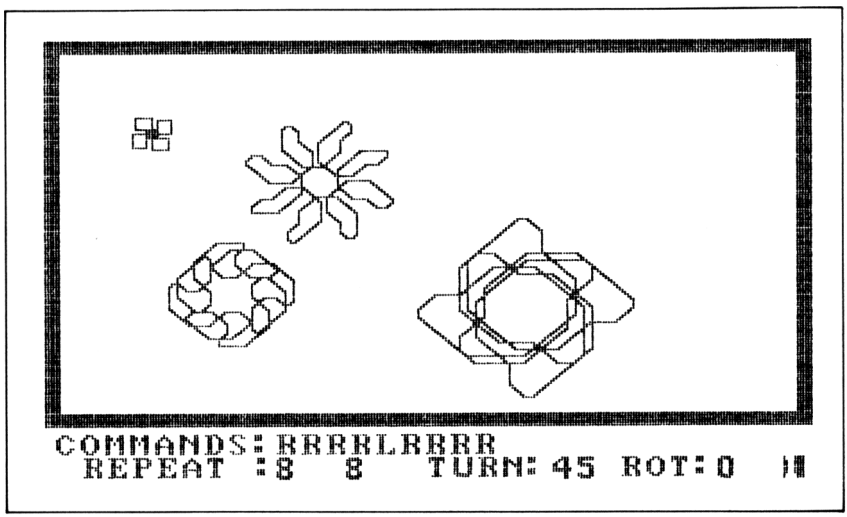


圖 3

程式

```
]LIST

10  REM **********************
20  REM **     SPIRALS !      **
30  REM **********************
35  DIM CM(30),KI(90),WD$(5)
45  DIM XI(15),YI(15)
50  REM ====POKE SOUND=====
52  FOR I = 0 TO 22: READ WW
54  POKE 768 + I,WW: NEXT I
55  DATA  169,10,133,1,173,48
56  DATA  192,136,208,4,198,1
```

```
57  DATA  240,8,202,208,246
58  DATA  166,0,76,4,3,96
60  REM  ----------------
70  REM    MAIN
80  REM  ----------------
85  GOSUB 500: GOSUB 600
90  GOSUB 700
100  REM  -------------
105  REM   READ COMMANDS
110  REM  -------------
115 ZZ =  PEEK (P2) - 135
120  XDRAW CU AT X,Y: XDRAW CU AT X,Y

125  IF ZZ < 0 OR ZZ > 217 THEN  GOTO
     115
130  IF KI(ZZ) = 0 THEN  GOTO 115
135  ON KI(ZZ) GOSUB 200,220,240,300,
     1100,390,380,370,475,150,150,150
     ,150
140  POKE P1,0: GOTO 115
150  REM  -- CHANGE START POS --
155 XA = XO + XI(KI(ZZ)):YA = YO + YI
     (KI(ZZ))
160  IF XA > 270 OR XA < 10 OR YA > 1
     60 OR YA < 10 THEN  RETURN
165  IF CS = 1 THEN  XDRAW CD AT XO,Y
     O
170 XO = XA:YO = YA: XDRAW CD AT XO,Y
     O:CS = 1
175  RETURN
200  REM  -- PRINT L/R ----
202  IF CC = K3 THEN  RETURN
206  IF (Y = 191 AND ZZ > 58) OR (Y =
     180 AND ZZ > 39 AND ZZ < 51) THEN
      RETURN
208 SV = ZZ + 135 - 159
```

```
210  HCOLOR= 0: DRAW CU AT X,Y
212  HCOLOR= 3: DRAW SV AT X,Y
214  IF Y = 191 THEN RN$(CC) =  CHR$
     (SV + 31): GOTO 218
216 CM(CC) = SV
218  GOSUB 220: RETURN
220  REM  -- R ARROW MOVE --
222  IF CC = K3 THEN  RETURN
224 X = X + 9
230 CC = CC + 1
235  RETURN
240  REM  -- L  ARROW MOVE --
245  IF CC = 1 THEN X = 85:Y = 371 -
     Y:K3 = 24 - K3: RETURN
250 X = X - 9
255 CC = CC - 1: RETURN
300  REM  -- COMPRESS ENTRY --
302  IF Y = 191 THEN  GOSUB 400: RETURN

304 X = 85:Y = 180
308  IF CC = 1 THEN  GOTO 315
310 CN = CC - 1:CC = 1
315  FOR I = 1 TO CN
320  HCOLOR= 0: DRAW CU AT X,Y
325  XDRAW CM(I) AT X,Y:X = X + 9
330 CM(CC) = CM(I):CC = CC + 1
335  NEXT :CC = CC - 1:CN = CC
340  FOR I = CN + 1 TO 21: HCOLOR= 0:
      DRAW CU AT X,Y:X = X + 9
345 CM(I) = 1: NEXT
349  GOSUB 350: RETURN
350  REM  - SET UP REPEAT READ -
355 X = 85:Y = 191
360  POKE P1,0:CC = 1
365 K3 = 3
368  RETURN
```

```
369  REM  -- SIZE CHANGE --
370 CO = 3:WD$(5) =  STR$ (SZ):LX = 2
    70:LY = 191:WD = 5: GOSUB 1000
371  CALL 768
372 CO = 0:SZ = SZ + 1: IF SZ > 5 THEN
    SZ = 1
373 WD$(5) =  STR$ (SZ):LX = 270:LY =
    191: GOSUB 1000
374 LN = SZ: RETURN
375  REM  --- ROTATION --
380 CO = 0:WD$(5) =  STR$ (RO):LX = 2
    45:LY = 191:WD = 5: GOSUB 1000
381  CALL 768
382 CO = 3:RO = 45 - RO
383 WD$(5) =  STR$ (RO):LX = 245:LY =
    191: GOSUB 1000
384 I3 = 1 - I3:I2 = I3
385  RETURN
389  REM  -- TURN DEGREE --
390 CO = 0:WD$(5) =  STR$ (TR):LX = 1
    85:LY = 191:WD = 5: GOSUB 1000
391  CALL 768
392 CO = 3:TR = 135 - TR
393 WD$(5) =  STR$ (TR):LX = 185:LY =
    191: GOSUB 1000
394 TS = 1 - TS
395  RETURN
400  REM  ------------
401  REM  DRAW ROUTINE
402  REM  ------------
404  IF CS = 1 THEN  XDRAW CD AT XO,Y
    O:CS = O
406  HCOLOR= 3: HPLOT XO,YO
408 RN =  VAL (RN$(1) + RN$(2))
410  FOR I = 1 TO RN
412  FOR J = 1 TO CN
```

```
414 S =  ABS (I1) +  ABS (I2):T1 = I1
    :T2 = I2
416  IF CM(J) = R AND TS = 1 THEN I1 =
    (T1 - T2) / S:I2 = (T1 + T2) / S
    : GOTO 424
418  IF CM(J) = R THEN T = I2:I2 = I1
    :I1 =  - T
420  IF CM(J) = L AND TS = 1 THEN I1 =
    (T1 + T2) / S:I2 = ( - T1 + T2) /
    S: GOTO 424
422  IF CM(J) = L THEN T = I1:I1 = I2
    :I2 =  - T
424  FOR K = 1 TO J
426 XO = XO + LN * I1:YO = YO + LN *
    I2
428  IF XO > 276 THEN XO = 276
430  IF XO < 1 THEN XO = 1
432  IF YO > 165 THEN YO = 165
434  IF YO < 0 THEN YO = 0
436  HPLOT  TO XO,YO
438  NEXT : NEXT
460 WD$(5) = "@@":WD = 5:CO = 0:LX =
    110:LY = 191: GOSUB 1000
462 WD$(5) =  STR$ (I):WD = 5:CO = 3:
    LX = 110: GOSUB 1000
464  HPLOT XO,YO
466  NEXT
468 X = 85:Y = 180:XO = 140:YO = 84:C
    C = 1
469 K3 = 21:I1 = 1:I2 = I3
470  RETURN
475  REM  - ERASE SCREEN -
480  HCOLOR= 0: FOR I = 1 TO 157: HPLOT
    7,5 + I TO 273,5 + I: NEXT :XO =
    140:YO = 84: XDRAW CD AT XO,YO
485 CS = 1: RETURN
```

```
500  REM  ------------
501  REM   INITIALIZE
502  REM  ------------
505 CD =  ASC (".") - 31
510 CU =  ASC ("@") - 31
515 CO = 3:LN = 1:TR = 90:RO = 0:SZ =
    1
520 I1 = 1:I2 = 0:I3 = 0:K3 = 21
525 CS = 1: SPEED= 255
530 CC = 1:R = 51:L = 45
535 XO = 140:YO = 84:X = 85:Y = 180
540 P1 =  - 16368:P2 =  - 16384
545 D$ =  CHR$ (13) +  CHR$ (4)
550  PRINT D$;"BLOAD ALPHA-TBL"
560  POKE 0,60: POKE 769,10
565  POKE 232,168: POKE 233,97
570  HGR2 : SCALE= 1: ROT= 0: HCOLOR=
    3
575  FOR I = 1 TO 25:CM(I) = 1: NEXT

599  RETURN
600  REM  -----------
605  REM   DATA
610  REM  -----------
615  DATA  69,75,25,41,42,43,44,45,46
    ,47,48,49,50,14,1,6,20,37,39,40,
    28,82,64,65,59
620  DATA  0,-4,-4,0,4,0,0,4
630  REM  - WORDS ---
635  DATA  "COMMANDS:","REPEAT :","TU
    RN:","ROT:"
650  REM  -- READ ARRAYS --
655  FOR I = 1 TO 13: READ K:KI(K) =
    1: NEXT
660  FOR I = 2 TO 13: READ K:KI(K) =
    I: NEXT
```

```
665  FOR I = 10 TO 13: READ XI(I),YI(
     I): NEXT
670  FOR I = 1 TO 4: READ WD$(I): NEXT

699  RETURN
700  REM  ----------
701  REM     SETUP
702  REM  ----------
705 CO = 3
710 XO = 140:YO = 84:LX = 278:LY = 16
     8
715  XDRAW CD AT XO,YO
720  GOSUB 800
730  FOR T = 1 TO 6: GOSUB 850: NEXT

740  REM  -- WORD STRINGS --
750 WD = 1:LX = 5:LY = 180: GOSUB 100
     0
755 WD = 2::LX = 15:LY = 190: GOSUB 1
     000
760 WD = 3:LX = 140:LY = 190: GOSUB 1
     000
765 WD = 4:LX = 210:LY = 190: GOSUB 1
     000
770 WD = 5:WD$(5) =  STR$ (TR):LX = 1
     85:LY = 191: GOSUB 1000
780 WD = 5:WD$(5) =  STR$ (RO):LX = 2
     45:LY = 191: GOSUB 1000
785  DRAW CU AT 270,191:CO = 0
790 WD = 5:WD$(5) =  STR$ (SZ):LX = 2
     70:LY = 191: GOSUB 1000
795 CO = 3: POKE P1,0
799  RETURN
800  REM  -- END POINTS --
810  HCOLOR= CO
820 XB = XO - LX / 2:XE = XO + LX / 2
```

```
830 YB = YO - LY / 2:YE = YO + LY / 2

840  RETURN
850  REM   ------ HOLLOW BOX -----
860  HPLOT XB,YB TO XE,YB TO XE,YE TO
     XB,YE TO XB,YB
880 XB = XB + 1:XE = XE - 1
882 YB = YB + 1:YE = YE - 1
890  RETURN
1000  REM   --- PRINT WORDS ---
1010  HCOLOR= CO
1020  FOR II = 1 TO  LEN (WD$(WD))
1030 SV =  ASC ( MID$ (WD$(WD),II,1))

1040  DRAW SV - 31 AT LX,LY
1050 LX = LX + 9: NEXT
1060  RETURN
1100  REM  ----- END -----
1110  IF  PEEK (800) <  > 1 THEN  HGR2
     : HOME : TEXT : HTAB 18: VTAB 10
     : PRINT "ENDED": END
1120  PRINT D$;"RUN MENU"
1200  REM  ================
```

變數使用說明

ZZ ……按鍵之ASCII 值減掉 128 ，當成KI 矩陣的足碼。

P1,P2 ……鍵盤及鍵盤選用（Key strobe）位址。

XO,YO ……若未使用Y、G、H、B鍵來改變繪圖起始位置，則XO，YO爲最初之繪圖起始點。

XA, YA ……繪圖起始點，（XA, YA)會因爲受Y、G、H、B鍵之移位而與（XO, YO)不同。

CD$……游標字形之字串。

CU$……■字形。

CS ……中心點旗號，決定是否在畫圖時將”·“游標由螢幕上消去。

X, Y……命令輸入時（R，L及空格），字形顯示之座標。

CC ……命令輸入時，字形之位置。

SV ……圖形號碼。

CN ……命令字形總數。

RN ……重覆之字數。

K3 ……命令輸入之字數限制（最多可輸入之字數）。

CO ……顏色值。

SZ ……放大倍數。

RO ……圖形整體旋轉之角度（0或45度）。

TR ……每一線段轉彎之角度（45 或 90 度）。

LN ……在DRAW副程式（400行）當中使用的放大因子，由SZ 來設定。

TS ……轉角旗號，當TS = 1 時，旋轉角度爲 45 度。

I1, I2, I3 ……控制左右轉之方向旗號。

T ……暫存用變數。

T1, T2 ……DRAW副 程式，旋轉45°時之暫存用變數。

S …… I1, I2 絕對值之和，在DRAW副程式中做45°轉彎時使用。

LX, LY ……印字副程式的起始位置。

R, L……“ R ”及“ L ”的ASCII 值。

D$ …… [RETURN] + CTRL – D 字串。

WD ……字形字串WD$ 的足碼。

矩陣使用說明

KI(90)……KI 之足碼爲按鍵之ASCII 值減掉135，其值則做爲ON KI(ZZ) GOSUB 之值支變數。

XI(15), YI(15) …(XO, YO)之步進值。

RN$(2)……重覆次數命令輸入時之暫存用矩陣變數。

CM(30)……記錄命令輸入字元之矩陣變數。

WD$(5)……字串陣列。

遊戲鍵說明

[→] ……將字形游標向右移動。

[←] ……將字形游標向左移動。

[ ] ……命令鍵：沿目前方向繪圖。

[R] ……命令鍵：右轉繪圖。

[L] ……命令鍵：左轉繪圖。

[RETURN] ……結束目前命令，改換他種命令，若目前所下達的是“重覆次數”命令（REPEAT）

，則便開始繪圖。

[#] ……清除圖框內圖案。

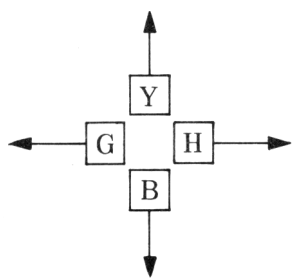
[/] ……改變放大倍數（1～5）。

[.] ……改變圖形整體旋轉角度（0或45度）。

[,] ……改變轉彎繪圖的角度（45或90度）。

[ESC] ……結束。

繪圖起始點之移動：



程式說明

10—50行 ……宣告維數。

60—90行 ……GOSUB 至各副程式做初始設定。

100—140行 ……等候按鍵，閃動游標。依所按字鍵至對應之副程式。

150—175行 ……Y、G、H、B鍵之處理副程式，移動游標，檢查是否超出邊界，在新位置畫出游標。

200—215行 ……接收轉彎命令及重覆次數之程式。（接收R、L、空格、及0～9等字鍵），將按

鍵之代表值存入 CM 矩陣，移游標至下一位置。

220—235行 ……字形游標右移程式，檢查是否超過最大字數的限制（轉彎命令最多 20 個字，重覆命令最多 2 位數）。將字形游標向右移一格，CC = CC + 1。

240—255行 ……字形游標左移程式，檢查是否目前已在最左端位置了，若是，則換行改變命令狀態。

300—349行 ……檢查是否在重覆命令輸入狀態？若是，則畫圖，若否，則將字形游標右邊之字形消除。

350—368行 ……重覆命令狀態之輸入初始設定。

370—374行 ……放大倍數設定：消去舊數目，計算並顯示新數目。

375—385行 ……旋轉角度設定：消去舊角度，計算並顯示新角度值。

390—395行 ……畫圖角度設定：消去舊角度，計算並顯示新角度值。

400—470行 ……繪圖副程式。

400—408行 ……若中央游標旗號（SW）= 1，則消去起始游標“・”：HPLOT 並且計算“RN”值。

410—438行 ……依“L、R及空格”命令來繪圖。

460—470行 ……顯示目前重覆次數。

475—485行 ……圖框內部的消除程式：清除圖框內部，再畫出圖形游標。

500—599行 ……初始設定，載入字形圖形表。

600—699行 ……讀入矩陣資料。

700—799行 ……畫出畫面、字幕之初始情況。

800—840行 ……計算方形之終點。

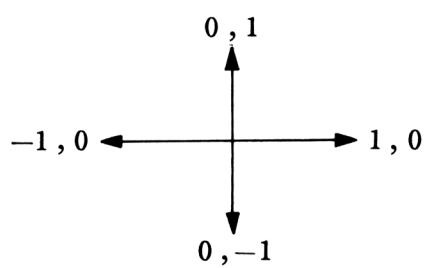
850—890行 ……畫方形副程式：畫出陸續縮小之方形，如果程式繼續執行，則將會把方形之內部填滿成實心圖形。

1000—1060行……顯示字串字形之副程式。

1100—1200行……結束程式，若是使用 menu 程式，則回到 menu 。

方向設定

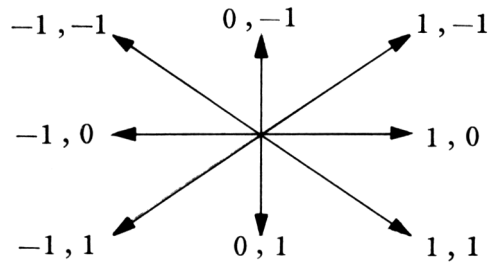
90°之畫面方位設定如下：



所以：A , B→B , -A （表示右轉）

A , B→-B , A （表示左轉）

45°之畫面方位設定如下：

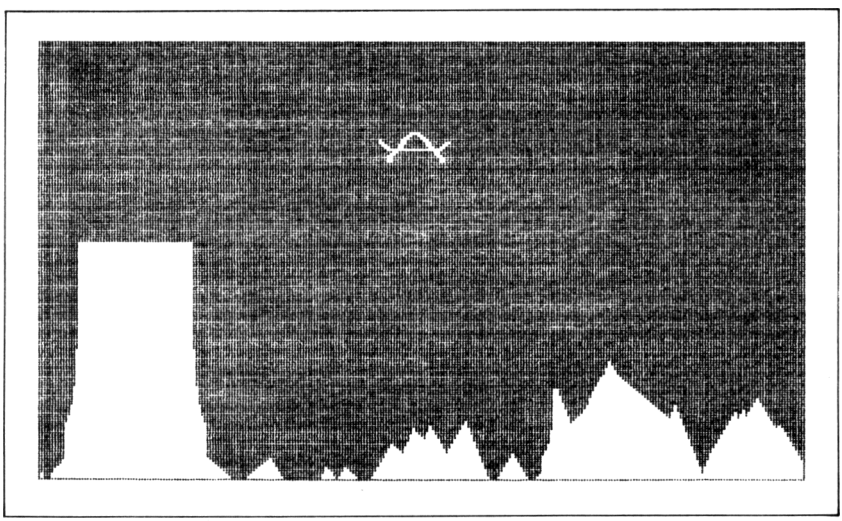


所以：A，B→（A−B）／ABS（A）+ABS（B），（A+B）／ABS（A）+ ABS（B）（右轉）

A，B→（A+B）／ABS（A）+ABS（B），（A−B）／ABS（A）+ ABS（B）（左轉）

第14章

外星人登陸(二)



遊戲說明

這是一個太空船登陸的與第八章不同的是，其繞捲的畫面效果。在這個Game 裡，駕駛員要學習降落在會移動的太空基地登陸台。此程式有7個不同的等級，等級愈高，登陸台愈小，你若安全降落則升級，否前降級。在最高的一個等級，您要降落在兩個垂直的懸壁上。若成功，則會給你一個意外的驚喜。假如你連

最低一級都沒有辦法安全降落，則你會遭到應得的處罰。

這個程式於高解析度X軸 100 點之前用亂數畫上崎嶇的陸地。行 640-1000 在作登陸台之建造。然後再用亂數畫個完整個畫面。您現在是太空船的駕駛員，您要指揮您的太空船安全降落。使用Ｉ，Ｊ，Ｋ，Ｍ鍵控制太空船的上下左右移動。注意按相同方向的鍵兩次會使您快速前進，按相反方向的鍵兩次會使您慢下來。

程式

```
]LIST

0  REM ************************
1  REM **     SPACE ROVER     **
2  REM **       GAME!!!       **
3  REM ************************
5  REM
70  LOMEM: 30000
80  TEXT : SPEED= 255: HOME : VTAB 1
     2: HTAB 14: FLASH : PRINT "  SP
     ACE  ROVER  ": NORMAL
90  REM   POKE IN 'WRAP AROUND'
100  DATA  169,49,133,29,198,29,198,
     29,164,29,185,72,80
110  DATA  133,26,136,185,72,80,133,
     30,162,8,134,28
120  DATA  198,28,166,28,189,120,80,
     24,101,30,133
130  DATA  27,160,0,177,26,72,160,1,
     177,26,136,145
```

```
140  DATA  26,200,200,192,40,208,245
     ,104,160,39,145
150  DATA  26,165,28,208,217,164,29,
     136,208,191,76
160  DATA  0,80,35,208,35,80,34,208,
     34,80,33,208
170  DATA  33,80,32,208,32,80,35,168
     ,35,40,34,168,34
180  DATA  40,33,168,33,40,32,168,32
     ,40,35,128,35
190  DATA  0,34,128,34,0,33,128,33,0
     ,32,128,32,0,0
200  DATA  4,8,12,16,20,24,28
210  FOR J = 0 TO 127
220  READ K: POKE 20480 + J,K
230  NEXT
250  REM  ====END OF POKING IN 'WRAP
      AROUND'====
290  DATA  1,0,4,0,46,46,46,46,45,37
     ,44,36,37,37,37
300  DATA 45,53,53,53,53,46,54,45,45
     ,44,44,44,148
310  DATA 210,219,51,46,46,54,29,199
     ,24,24,8,1,24
320  DATA 63,63,63,63,63,63,63,54,55
     ,55,126,5,0
330  FOR J =     0 TO 53
340  READ K
350  POKE 25000 + J,K
360  NEXT J
370  POKE 232,168: POKE 233,97: REM
     ---LOCATION OF SHAPE TABLE IS $
     61A8---
390  REM ====END OF POKING IN SHAPE
     TABLE=====
420  DEF  FN R(N) =  INT ( RND (1) *
```

```
     N) + 1: REM ---RETURN A RANDOM
     NUMBER FROM 1 TO X---
430  DIM LND(279),RNK$(J)
435  FOR J = 1 TO 7: READ RNK$(J): NEXT

440  TEXT
450  GOSUB 2500
455  IF LEVEL = 1 THEN  GOSUB 2000
458 ST = 0:AL = 1
459 PLIM = 100:PASS = 0: IF LEVEL >
     2 THEN PLIM =  INT (21 / LEVEL)

460  POKE 20481,27
470  POKE 20549,96: REM   INSERT "RT
     S"
480 MDPT = 151:LEVEL =  INT (LEVEL)
490  HGR : POKE  - 16302,0: REM ---F
     ULL SCREEN---
500  HCOLOR= 3
510 SIGN = 1
520 HGT = MDPT: HPLOT 0,0 TO 0,HGT:L
     ND(0) = HGT
530 CH = 3:LIMIT = 116
540  FOR J = 1 TO 100
550  IF  FN R(4) <  > 1 THEN 580
560 SIGN =  FN R(2): IF SIGN = 2 THEN
     SIGN =  - 1
570 CH = SIGN *  FN R(3)
580 HGT = HGT + CH
590  IF HGT < LIMIT THEN HGT = LIMIT

600  IF HGT > 191 THEN HGT = 191
610  HPLOT J,0 TO J,HGT
620 LND(J) = HGT
630  NEXT J
640 LAST = HGT:LIMIT = 87
```

```
650 K = 6
660  FOR J = 101 TO 106
670 LND(J) = LAST - (LAST - 87) / K:
     HPLOT J,0 TO J,LND(J)
680 K = K - 1: NEXT J
690  IF LEVEL > 2 THEN 790
700 SPAN = 4 / LEVEL * 21
710  FOR J = 106 TO 106 + SPAN
720 LND(J) = 87: HPLOT J,0 TO J,LND(
    J)
730  NEXT J
740 K = 2
750  FOR J = 106 + SPAN TO 111 + SPA
    N
760 LND(J) = LAST - (LAST - 87) / K:
     HPLOT J,0 TO J,LND(J)
770 K = K + 1: NEXT J
780  GOTO 1010
790  REM ====FORM A CREVICE====
800  IF LEVEL = 3 THEN SPAN = 29:CRE
    VICE = 6
810  IF LEVEL = 4 THEN SPAN = 19:CRE
    VICE = 8
820  IF LEVEL = 5 THEN SPAN = 9:CREV
    ICE = 10
830  IF LEVEL = 6 THEN SPAN = 3:CREV
    ICE = 16
840  IF LEVEL = 7 THEN SPAN = 1:CREV
    ICE = 17
850  FOR J = 106 TO 106 + SPAN
860 LND(J) = 87: HPLOT J,0 TO J,LND(
    J)
870  NEXT J
880 K = 1
890  FOR J = 106 + SPAN TO 106 + SPA
    N +  INT (CREVICE / 2)
```

```
900 LND(J) = LIMIT + K: HPLOT J,0 TO
    J,LND(J):K = K + 1
910  NEXT J:K = K - 2
920  FOR J = 107 + SPAN +  INT (CREV
    ICE / 2) TO 106 + SPAN + CREVIC
    E
930 LND(J) = LIMIT + K: HPLOT J,0 TO
    J,LND(J):K = K - 1
940  NEXT J
950  IF LEVEL = 7 THEN SUB = 1: GOTO
    970
960 SUB = 0
970  FOR J = 107 + SPAN + CREVICE TO
    106 + 2 * SPAN + CREVICE - SUB
980 LND(J) = 87: HPLOT J,0 TO J,LND(
    J)
990  NEXT J
1000 SPAN = 1 + 2 * SPAN + CREVICE: GOTO
    740
1010 LIMIT = 101
1020 HGT = LAST
1030 CH = 3:LIMIT = 116
1040  FOR J = 111 + SPAN TO 274
1050  IF  FN R(4) <  > 1 THEN 1080
1060 SIGN =  FN R(2): IF SIGN = 2 THEN
    SIGN =  - 1
1070 CH = SIGN *  FN R(3)
1080 HGT = HGT + CH
1090  IF HGT < LIMIT THEN HGT = LIMI
    T
1100  IF HGT > 191 THEN HGT = 191
1110  HPLOT J,0 TO J,HGT
1120 LND(J) = HGT
1130  NEXT J
1140 LIM = LIM + 4
1150 LAST = HGT
```

```
1160 DF = MDPT - HGT:K = 5:C = 39
1170  FOR J = 275 TO 279
1180 HGT = LAST + DF / K: HPLOT J,0 TO
     J,HGT
1190 LND(J) = HGT
1200 K = K - 1
1210  NEXT J
1220  REM ====FINISHED DRAWING LAND=
     ===
1230 LIMIT = 87
1240  SCALE= 1: ROT= 0:X = 139:C = 1

1250 XS =  - 1:YS = 0: REM ---SET IN
     ITIAL SPEED---
1260 Y = LIMIT / 2: REM    Y-COORDIN
     ATE
1270  HCOLOR= 0: IF XS = 0 AND YS =
     0 THEN  DRAW 1 AT X,Y: PRINT ""
     :ST = ST + 1: GOTO 1350
1280 XOLD = X:X = X + XS:YOLD = Y:Y =
     Y + YS
1290  IF X < 0 THEN X = 280 + X:PASS
      = PASS + 1: IF PASS > PLIM THEN
     X = XOLD:Y = YOLD:M$ = "TOO MAN
     Y PASSES": GOTO 1490
1300  IF Y < 3 THEN Y = 3:YS = 0
1310  IF X > 279 THEN X = X - 280
1320  IF Y > 78 THEN AL = 0: GOTO 14
     30
1330  DRAW 1 AT X,Y
1340  HCOLOR= 3: DRAW 1 AT XOLD,YOLD

1350  CALL 20480:C = C + 1:QQ =  PEEK
     ( - 16336): IF C = 40 THEN C =
     0
1355  IF ST =  INT (21 / LEVEL) THEN
```

```
     M$ = "SHIP NOT MOVING": GOTO 14
     90
1360 K =  PEEK ( - 16384): POKE  - 1
     6368,0: IF K > 127 THEN 1380
1380  IF K = 202 THEN XS = XS - 1
1390  IF K = 203 THEN XS = XS + 1
1400  IF K = 201 THEN YS = YS - 1
1410  IF K = 205 THEN YS = YS + 1
1420  GOTO 1270
1430 PLACE = C * 7 + X + 3: IF PLACE
      > 279 THEN PLACE = PLACE - 280

1440  CALL 20480: DRAW 1 AT X,Y: HCOLOR=
     3: DRAW 1 AT XOLD,YOLD
1450  IF LND(PLACE) <  > LIM THEN 14
     90
1460 PLACE = PLACE + 19: IF PLACE >
     279 THEN PLACE = PLACE - 280
1470  IF LND(PLACE) <  > LIM THEN 14
     90
1480  TEXT : HOME : FLASH : VTAB 12:
      HTAB 9: PRINT "SUCCESSFUL LAND
     ING": NORMAL : FOR P = 1 TO 600
     : NEXT P: POKE  - 16304,0:GOOD =
     GOOD + 1:SUC = 1:M$ = "SOFT LAN
     DING": GOTO 1500
1490  TEXT : HOME : VTAB 12: HTAB 15
     : PRINT "C R A S H !!!": FOR P =
     1 TO 600: NEXT P: POKE  - 16304
     ,0:BAD = BAD + 1:SUC = 0: IF AL
      = 0 THEN M$ = "MISSD!"
1491  HCOLOR= 3: FOR P = 1 TO 500: NEXT
     P: DRAW 1 AT X,Y:R = 0
1494  FOR YN = Y TO 191 STEP 3:R = R
      + 1: ROT= R: XDRAW 1 AT X,YN -
     3
```

```
1496  NEXT
1498  FOR I = 0 TO 2: PRINT  CHR$ (7
      ): FOR J = 0 TO 10:QQ =  PEEK (
       - 16336): NEXT J: NEXT I
1500  FOR I = 0 TO 3: PRINT  CHR$ (7
      ): NEXT I: POKE  - 16368,0: GET
      A$
1510  TEXT : HOME
1520  IF SUC = 1 THEN LEVEL = LEVEL +
      1:MIS = 0
1530  IF LEVEL = 8 THEN 1700
1540  IF SUC = 0 THEN MIS = MIS + 1:
       IF MIS = 2 THEN LEVEL = LEVEL -
      1:MIS = 0
1550  IF LEVEL = 0 THEN 1800
1560  PRINT "RANK AFTER MISSION: ";R
      NK$(LEVEL)
1570  PRINT : PRINT "SUCCESSFUL LAND
      INGS = ";GOOD
1580  PRINT "CRASHES"; SPC( 13);"= "
      ;BAD
1581  FLASH
1582  VTAB 9: IF SUC = 0 AND MIS = 0
       THEN  HTAB 14: PRINT "DEMOTED!
      !": NORMAL
1583  IF SUC = 1 THEN  HTAB 8: PRINT
      "P R O M O T E D !!!"
1584  NORMAL
1585  VTAB 12: PRINT "LEVELS:": PRINT
      : FOR J = 1 TO 7: PRINT J;") ";
      RNK$(J): NEXT
1586  VTAB 7: PRINT "REPORT: ";: FLASH
      : PRINT M$: NORMAL : VTAB 21
1590  PRINT : PRINT "PRESS A KEY FOR
       THE NEXT MISSION ";: POKE  - 1
      6368,0: GET A$: GOTO 458
```

```
1700  REM  WINNER
1710  HOME : PRINT "CONGRATULATION!!
      YOU HAVE SUCCESSFULLY  COMPLET
     ED THE SEVENTH AND MOST DIFFICU
     LTLEVEL OF THE COURSE.": PRINT

1720  PRINT "HERE ARE THE STATISTICS
      OF YOUR TRIALS"
1730  PRINT : PRINT : PRINT "LANDING
     S = ";GOOD
1740  PRINT "CRASHES = ";BAD
1750  PRINT : PRINT "TO DESIGN YOUR
     OWN MODULES, MODIFY LINES 790 -
      840."
1760  END
1800  REM  LOSER
1810  HOME
1820  PRINT "YOU HAVE UTTERLY FAILED
      IN YOUR ATTEMPTSTO LAND THE SH
     IP!"
1830  PRINT : PRINT "HERE ARE YOUR S
     TATISTICS": PRINT
1840  PRINT "LANDINGS = ";GOOD
1850  PRINT "CRASHES  = ";BAD
1860  PRINT "YOU ARE A DISGRACE BE S
     HOT  AT THE CRACK OF DAWN!"
1870  END
2000  REM       INSTRUCTION
2010  HOME : SPEED= 200
2020  VTAB 1: FOR J = 1 TO 40: HTAB
     J: PRINT "*";: NEXT J
2030  FOR J = 1 TO 23: HTAB 40: VTAB
     J: PRINT "*";: NEXT J
2040  VTAB 22: FOR J = 40 TO 1 STEP
      - 1: HTAB J: PRINT "*";: NEXT
     J
```

```
2050  FOR J = 23 TO 1 STEP  - 1: HTAB
      1: VTAB J: PRINT "*";: NEXT J
2060  POKE 32,2: POKE 33,37: POKE 34
      ,1: POKE 35,22: SPEED= 100
2070  VTAB 3: PRINT  TAB( 11);"SPECI
      AL REPORT": PRINT
2080  FOR P = 1 TO 100: NEXT P
2090  PRINT "DEAR CADET:"
2100  HTAB 2: PRINT "CONGRATULATIONS
      ! YOU'RE GOING TO "
2110  PRINT "LEARN HOW TO MANEUVER T
      HE ";: INVERSE : PRINT "SPACE R
      OVER";: NORMAL
2120  PRINT "SPACE SHIP.  THE CONTRO
      LS ARE THE"
2130  PRINT "'I', 'J', 'K', AND 'M'
      KEYS WHICH"
2140  PRINT "MOVE THE SHIP IN THE RE
      SPECTIVE"
2150  PRINT "DIRECTIONS."
2160  HTAB 2: PRINT "WITH THESE CONT
      ROLS IT IS YOUR JOB"
2170  PRINT "TO LAND ON THE MOVING P
      LATFORM.
2180  PRINT : PRINT  TAB( 15);"CAUTI
      ONS"
2190  PRINT "1) KEEP MOVING AT ALL T
      IMES"
2200  PRINT "2) NEVER GO BELOW THE P
      LATFORM"
2210  PRINT : PRINT "PRESS A KEY TO
      BEGIN";: POKE  - 16368,0: GET A
      $
2220  SPEED= 255: RETURN
2500  HOME : REM   CHOOSE A LEVEL
2510  PRINT "CHOSE YOUR STARTING LEV
```

```
     EL..."
2520  VTAB 5
2530  FOR J = 1 TO 7
2540  PRINT J;") ";RNK$(J): PRINT
2550  NEXT J
2600  PRINT : INPUT "CHOICE==>";LEVE
     L
2610  IF LEVEL < 1 OR LEVEL > 7 THEN
     2500
2620  RETURN
2640  PRINT J;") ";RNK$(J): PRINT
5000  DATA  PRIVATE, PRIVATE SECOND
     CLASS,CORPORAL,SERGEANT,SECOND
     LIEUTENANT,FIRST LIEUTENANT,CAP
     T. ZORRO
```

圖號	1
圖案	

圖案及其圖形號碼

```
*61A8.61DD

61A8- 01 00 04 00 2E 2E 2E 2E
61B0- 2D 25 2C 24 25 25 25 2D
61B8- 35 35 35 35 2E 36 2D 2D
61C0- 2C 2C 2C 94 D2 DB 33 2E
61C8- 2E 36 1D C7 18 18 08 01
61D0- 18 3F 3F 3F 3F 3F 3F 3F
61D8- 36 37 37 7E 05 00
```

圖案圖形表

繞捲效果說明

本程式的特色是在於其繞捲的畫面，但是APPLE 的高解像畫面共使用了8 K的記憶位址，如果要造出繞捲的效果，則勢必要將8 K的位址移動，以ABSIC 語言來執行的話，是要花費相當長時間的，所以我們便借用快速的機械語言來輔助處理。或許您已經注意到BASIC 的易讀性和機械語言的快速，利用此種二合一的技巧，把簡單但要求快速的工作讓機器語言來做，如此可使程式編寫容易，且執行速度快。我們設計出來的繞捲程式，如下所示：

```
*5000LLLL

5000-    LDA   #$1B       ; 欲旋轉行數乘 2 加 1
5002-    STA   $1D
5004-    DEC   $1D
5006-    DEC   $1D
5008-    LDY   $1D
500A-    LDA   $5048,Y    ; 取位址低位元組
500D-    STA   $1A
500F-    DEY
5010-    LDA   $5048,Y    ; 取位址高位元組
5013-    STA   $1E
5015-    LDX   #$08
5017-    STX   $1C
5019-    DEC   $1C
501B-    LDX   $1C
501D-    LDA   $5078,X    ; 取 box 裡之差值
5020-    CLC
5021-    ADC   $1E
5023-    STA   $1B
5025-    LDY   #$00
```

```
5027-   LDA   ($1A),Y   ; 取出第一個資料
5029-   PHA              ; 存起來
502A-   LDY   #$01
502C-   LDA   ($1A),Y   ; 移動資料
502E-   DEY
502F-   STA   ($1A),Y
5031-   INY
5032-   INY
5033-   CPY   #$28       ; 移完 40 個?
5035-   BNE   $502C      ; 否，至$502C
5037-   PLA              ; 把第一個資料取出
5038-   LDY   #$27
503A-   STA   ($1A),Y   ; 存於最後
503C-   LDA   $1C
503E-   BNE   $5019      ; 移完一個 box?
5040-   LDY   $1D
5042-   DEY
5043-   BNE   $5004      ; 全部移完?
5045-   RTS   (JMP)
5046-         (00)
5047-         (50)
```

使高解像畫面繞捲的機械語言

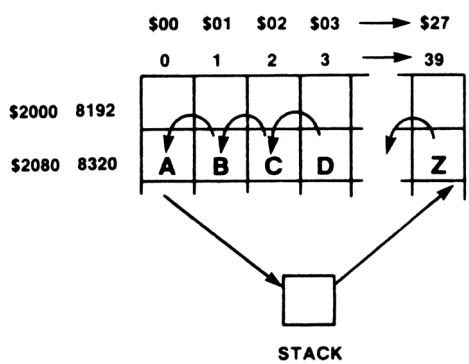
```
*5048.5080

5048- 23 D0 23 50 22 D0 22 50
5050- 21 D0 21 50 20 D0 20 50
5058- 23 A8 23 28 22 A8 22 28
5060- 21 A8 21 28 20 A8 20 28
5068- 23 80 23 00 22 80 22 00
5070- 21 80 21 00 20 80 20 00
5078- 00 04 08 0C 10 14 18 1C
5080- 00
*($5048～5077是位址表，$5078～$5080是box內的相對位址)
```

高解像畫面水平線的起始位址表

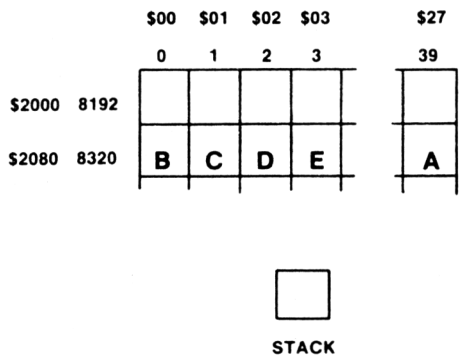
---

*有關高解像畫面起始位址表的徹底使用，請參考Ref.7

這個機械語言程式存於由$5000 （十進位20480)開始的位址。要把高解像度整個畫面向左旋轉，請參看圖一，第一列是$00……$27，第二列0～39 是高解像度每一線（ 0～279點）所佔的記憶位址。假設圖一是原本的情況，我們可看到位址8320,8321,……內的值是A ，B……，若要向左旋轉，那怎作呢？我想您已知道答案，就是如圖一箭頭所示的把B搬到8320，把C搬到8321 ，這不是很簡單嗎？對，就是這麼簡單！這就是旋轉程式的動作原理。但是我們再想想若只照這方法一直左移，那最後8320及其後位址的內含值不都變成了Z嗎？就是這關鍵所在了。再看圖一，下方有一STACK 這個方塊。我們利用這個方塊，把8320的內含值A先存到STACK ，然後再把B移至3820 ，C移至8321 ，……，然後再把STACK 的內含值移至8359 ，就是移到第 39 個位址。經過了這樣的手續，即成了圖二所示的結果。比較圖一和圖二，您是否看出了ABCD…Z已變成BCDE…A ，這不就是我們的目的"旋轉"了嗎？上述這個觀念很重要，是設計GAME的必備知識。



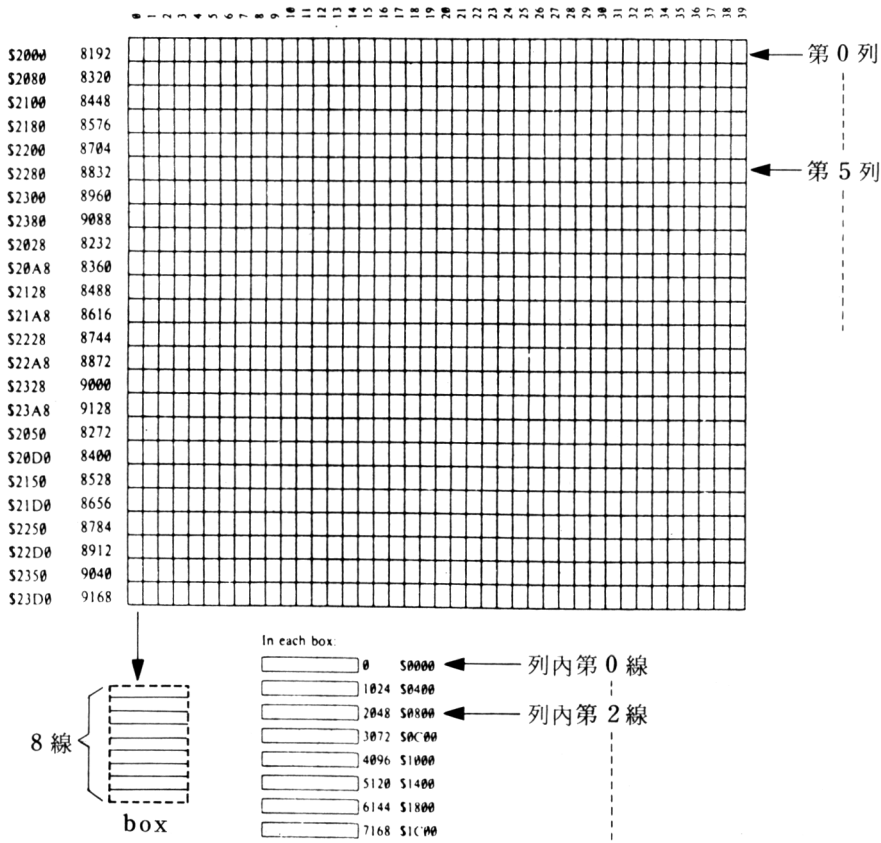
圖一　位移前狀況



圖二　位移後狀況

剛才所說的旋轉 1 行，是高解析度水平線上每一點的水平旋轉，但是畫面共有 191 線，即我們要作 192 次旋轉，乍看之下眞是很大的工程，但是事實上只要 1／192 即可作好這些工作。因爲 192 次都是作同樣的工作，我們可以作一迴路即可。我想您可以想像如果用 BASIC 來作的話，是多麼的耗時間了吧！但用機械語言來作這些工作，則既輕鬆又快速，這樣即可以達到畫面的特殊效果了。

最後來說明這個使高解像畫面旋轉程式，請參照圖三的APP-LE 高解析度記憶位址對應圖：



圖上有 40 行（ 0～39 ） 24 列，列的位址由$2000 至$23D0，每一列裡又含有 8 條水平線 24*8=192 ，即Y軸有 24 列 192 線，這是大家都知道的，但是這列內的 8 條水平線的位址是多少呢！在圖的下方有這樣的英文字“ In each box”;就是這裡，在 box 的右方印有 8 個數字：$0000,$0400,$0800

,$0C00,$1000,$1400,$1800,$1C00,所以高解像畫面上每一水平線的起始(左端)位址算法如下:

例:第0線的位址=$2000+$0000=$2000。

第1線的位址=$2000+$0400=$2400。

第 100 線的位址=$21A8(第 11 列,因一行有8線)+$0C00(第3線)。

在我們的機械語言旋轉程式裡,先設定好Y軸每一列的位址和In each box的位址,然後利用上述原理算出Y軸每一水平線的位址,再利用我們談到的旋轉觀念來使畫面旋轉。這個旋轉程式您可用到任何高解像畫面的場合,想旋轉時,只要在程式裡使用

CALL 20480

便可造出向左繞捲效果來。如果你用

POKE 20549,96

將$5045 位址內的$4C(即JMP)改成$60(即RTS, 10 進位值爲 96),則CALL 20480的結果是畫面向左移動一個位元組(7點),而不是不停的繞捲,我們的GAME便是這樣作的。

程式說明

這是一個有趣的GAME。除了前述的繞捲效果之外,您可能對程式如何檢查降落成功與否也感到興趣吧?這是用一個計數變

---

*因第100線爲12*8+4,但由0算起,所以是(11+1)*8+(3+1)。

數和陣列變數來作的，當在用亂數畫陸地的時候同時把X軸每一點（0～279）的陸地高度（Y軸），存在LND(X)陣列變數內。每旋轉畫面一次計數變數即加一，但被限制於40以內（行1350），由1430行即可知道太空船所對應的陸地位址。行中C爲計數用，X爲太空船的X軸位址。C＊7是因X軸每行有40個位元組，即高解析度的280點，可知其中一個位元組共包含了7點，故C乘以7。

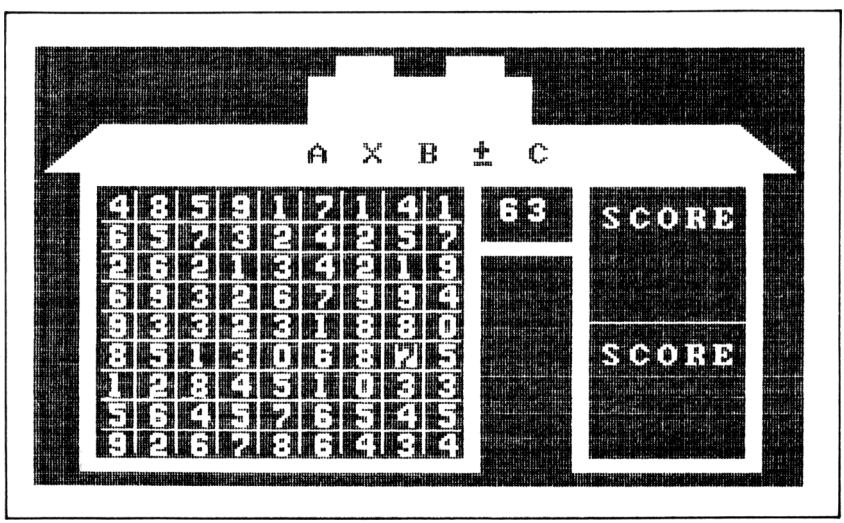
登陸台的高度是一定的。利用行1430的公式即可知道太空船所在位址底下的陸地高度。所以當太空船降落時只要判斷陸地的高度即知道太空船是否安全降落。

由於本程式的圖形表及旋轉程式均不是很長，所以不必另外建立機械語言的檔案，只要在BASIC 主程式中將圖形表及旋轉程式POKE進來使用，就可以了。程式中 100 行～ 250 行便是建立旋轉程式用的，而 290 行～ 360 行則是用來建立圖形表。

另外要提到的是，旋轉程式中$5001 的內含值是存放所欲旋轉的列數資料，其值等於（列數）＊ 2+1 ，由圖三我們可以看出畫面共分成水平 24 列（每列內含8線），如果我們要令畫面底下的5列移動，$5001 便要存入 5＊2+1=11=$0B，程式中的第 460 行就是用這種方法，現在你應該知道它到底旋轉多少行了吧！不過要注意這裡所說的5列是由底下往上算的。

第15章

精打細算



遊戲說明

這是一個算術遊戲，一開始，畫面上有一個 9×9 共 81 個方格的字盤，字盤上以亂數方式填入數字，接著電腦會給你一答案，你必須在限定的時間內在字盤上找出一組以水平或垂直或45°直線相連的 3 個數，這 3 個數以 A×B±C 的公式運算的結果必順以答案相等，如此這個答案便加到你的分數上了。譬如，現

在電腦顯示的答案是 48 ，你可以找出下列 3 個一組的數字：

$6 \times 8 \pm 0 = 48$

$6 \times 7 + 6 = 48$

$7 \times 8 - 8 = 48$

.....

如果找到了，你便得到 48 分。

程式一執行，你必須按下 RETURN 鍵，才能令遊戲開始，遊戲可以由一個人或兩個人同時進行比賽，每人有一個可控制的游標，轉動旋鈕 * (Paddle)，你可以選擇以目前游標位置爲中心的 3 個數目（水平或垂直或±45°），或是轉動旋鈕使游標不閃爍，這時按下遊戲按鈕，便可以移動游標到你所選擇那組數目字的位置了。這個遊戲不只是在於誰搶先找到正確的一組數字，並且也要靈活的控制旋鈕及按鈕，由於APPLE 原來的遊戲控制器是以一種稱爲Paddle 的單一可變電阻旋鈕構成的，而本程式也是依此種旋鈕而設計的，所以用一般常見的搖桿來控制，可能更加困難，不過沒關係，只要你得分超過 100 便是贏了。

又如果是單獨一個人玩這個遊戲，也不可以太慢於思考，只要作答時間超過，APPLE 就又會變換了一個新數字。

程式

```
JLIST

10  REM  **********************
20  REM  **     EQUATE!!!     **
30  REM  **     A X B = C     **
40  REM  **********************
45  DIM NM(100)
50  REM  ====POKE SOUND====
52  FOR I = 0 TO 22: READ WW
54  POKE 768 + I,WW: NEXT I
55  DATA  169,10,133,1,173,48
56  DATA  192,136,208,4,198,1
57  DATA  240,8,202,208,246
58  DATA  166,0,76,4,3,96
60  REM  ---------------
70  REM        MAIN
80  REM  ---------------
85  GOSUB 500: GOSUB 600
90  GOSUB 700: GOSUB 400
95  GOTO 100
100  REM  -------------
101  REM   READ KEY
102  REM  -------------
105  FOR PN = 0 TO 1:Q =  PDL (PN): GOSUB
     120
110  GOSUB 350
115  IF  PEEK (PB(PN)) > 127 THEN  GOSUB
     250
119  NEXT : GOTO 105
120  REM  -- COMPUTE PADDLE --
125 K(PN) =  INT ((Q + 44) / 50) + 1
```

```
130  IF K(PN) > 1 AND K(PN) < 6 THEN
     : GOSUB 140: RETURN
135 DS(PN) = 1: RETURN
140  REM   -- SHOW 3-IN-ROW --
141  IF (PX(PN) = 28 OR PX(PN) = 148
     ) AND (PY(PN) = 74 OR PY(PN) =
     178) THEN DS(PN) = 0: RETURN
142  IF (PX(PN) = 28 OR PX(PN) = 148
     ) AND K(PN) <  > 4 THEN DS(PN) =
     0: RETURN
143  IF (PY(PN) = 74 OR PY(PN) = 178
     ) AND K(PN) <  > 2 THEN DS(PN) =
     0: RETURN
144 DS(PN) = 1
145  FOR TT = 1 TO 2
146  XDRAW  ASC ("④") - 31 AT PX(PN)
      + X1(K(PN)),PY(PN) + Y1(K(PN))

147  XDRAW  ASC ("④") - 31 AT PX(PN)
     ,PY(PN)
148  XDRAW  ASC ("④") - 31 AT PX(PN)
      + X2(K(PN)),PY(PN) + Y2(K(PN))

149  NEXT : RETURN
250  REM  -- MOVE ROUTINE --
255  IF DS(PN) = 0 THEN  RETURN
260  ON K(PN) GOSUB 270,310,320,330,
     340,280
269  RETURN
270  REM   -- HORZ  MOVE --
271  XDRAW  ASC ("④") - 31 AT PX(PN)
     ,PY(PN)
272 PX(PN) = PX(PN) + 15:PS(PN) = PS
     (PN) + 1
274  IF PS(PN) > 81 THEN PX(PN) = 28
     :PY(PN) = 74:PS(PN) = 1
```

```
276  IF PX(PN) > 148 THEN PX(PN) = 2
     8:PY(PN) = PY(PN) + 13
278  XDRAW  ASC ("@") - 31 AT PX(PN)
     ,PY(PN)
279  POKE 0,5 + 200 * PN: CALL 768: RETURN

280  REM   -- VERT MOVE ---
282  XDRAW  ASC ("@") - 31 AT PX(PN)
     ,PY(PN)
284 PY(PN) = PY(PN) + 13:PS(PN) = PS
     (PN) + 9
286  IF PY(PN) > 180 THEN PX(PN) = P
     X(PN) + 15:PY(PN) = 74:PS(PN) =
      ABS (80 - PS(PN)): IF PS(PN) =
     10 THEN PX(PN) = 28:PY(PN) = 74
     :PS(PN) = 1
288  XDRAW  ASC ("@") - 31 AT PX(PN)
     ,PY(PN)
289  POKE 0,5 + 200 * PN: CALL 768: RETURN

300  REM   -- GET 3-IN-ROW --
310 A = NM(PS(PN) - 1):B = NM(PS(PN)
     ):C = NM(PS(PN) + 1): GOSUB 345
     : RETURN
320 A = NM(PS(PN) - (9 + 1)):B = NM(
     PS(PN)):C = NM(PS(PN) + (9 + 1)
     ): GOSUB 345: RETURN
330 A = NM(PS(PN) - 9):B = NM(PS(PN)
     ):C = NM(PS(PN) + 9): GOSUB 345
     : RETURN
340 A = NM(PS(PN) - (9 - 1)):B = NM(
     PS(PN)):C = NM(PS(PN) + (9 - 1)
     ): GOSUB 345: RETURN
344  REM  -- COMPUTE ANSWER --
345 A1 = A * B - C:A2 = A * B + C:A3
      = B * C - A:A4 = B * C + A
```

```
346  IF AN = A1 OR AN = A2 OR AN = A
     3 OR AN = A4 THEN : FOR TT = 1 TO
     5: POKE 0,TT * 20: POKE 769,50:
      CALL 768: NEXT : POKE 769,2: GOSUB
     375: GOTO 349
348  POKE 0,250: POKE 769,100: CALL
     768: POKE 769,2
349  RETURN
350  REM  - SHOW ANS --
352 TI = TI - 1
354  IF TI > 0 THEN  RETURN
360 CO = 3:NX = 170:NY = 75:SC = AN:
      GOSUB 1000
362 AN =  INT ( RND (1) * 90) + 1
364 CO = 0:SC = AN:NX = 170: GOSUB 1
     000
366  FOR TT = 1 TO 10: POKE 0,5 * TT
     : CALL 768: NEXT :TI = 600: RETURN
375  REM  -- SCORE --
380 CO = 3:NX = NX(PN):NY = NY(PN):S
     C = SC(PN): GOSUB 1000
385 SC(PN) = SC(PN) + AN
390 CO = 0:NX = NX(PN):NY = NY(PN):S
     C = SC(PN): GOSUB 1000
395  IF SC(PN) > SL THEN  GOTO 1100
399 TI = 0: FOR T = 1 TO 100: GOSUB
     350: RETURN
400  REM  --START--
401  IF EN = 1 THEN  GOTO 406
402  GOSUB 410
406  GOSUB 450:CO = 0: GOSUB 475: RETURN

408  REM  -- WAIT INPUT -
410 CO = 0: GOSUB 490
412 CO = 3:WD = 2:LX = 30:LY = 46: GOSUB
     950
```

```
414 WD = 3:LX = 30:LY = 56: GOSUB 95
    0
415  REM  --LOOP --
418  IF  PEEK (P2) = 155 THEN  GOTO
    1150
420  IF  PEEK ( - 16384) > 128 THEN
     GOTO 415
422  FOR T = 1 TO 40: NEXT
424  IF  PEEK ( - 16384) > 128 THEN
    430
428  GOTO 415
430 CO = 0:WD = 2:LX = 30:LY = 46: GOSUB
    950
438 CO = 0:WD = 3:LX = 30:LY = 56: GOSUB
    950
439  RETURN
450  REM  -- MIX NUMBERS --
455  FOR I = 81 TO 2 STEP  - 1
460 J =  INT ( RND (1) * I) + 1:T =
    NM(I)
465 NM(I) = NM(J):NM(J) = T: NEXT
474  RETURN
475  REM  -- PLACE NUMBERS --
476 CO = 3: GOSUB 490:CO = 0
478 NX = 13:NY = 74:J = 0
480  HCOLOR= CO: FOR K = 1 TO 9: FOR
    I = 1 TO 9
482 J = J + 1:NX = NX + 15
484 NM$ =  STR$ (NM(J))
486  DRAW  ASC (NM$) - 31 AT NX,NY
488  NEXT :NX = 13:NY = NY + 13: NEXT

489  RETURN
490  REM  --DISPLAY EQUATION --
492 WD = 4:LX = 100:LY = 50: GOSUB 9
    50
```

```
494 WD = 5:LX = 160:LY = 49: GOSUB 9
    50:WD = 6:LY = 55:LX = 160: GOSUB
    950
495  RETURN
500  REM  ------------
501  REM    INITIALIZE
502  REM  ------------
505  SPEED= 255
510 PN = 0:SL = 100
515 PS(0) = 41:PS(1) = 41
520 PX(0) = 88:PX(1) = 88
525 PY(0) = 126:PY(1) = 126
530 P1 =  - 16368:P2 =  - 16384:PB(0
    ) =  - 16287:PB(1) =  - 16286
535  POKE P1,0
540 D$ =  CHR$ (13) +  CHR$ (4)
550  PRINT D$;"BLOAD ALPHA-TBL"
560  IF  PEEK (800) <  > 1 THEN R =
     RND ( -  PEEK (78) -  PEEK (79
    ) * 256)
565  POKE 0,100: POKE 769,2
570  POKE 232,168: POKE 233,97
580  HGR2 : SCALE= 1: ROT= 0: HCOLOR=
    3
599  RETURN
600  REM  ----------
601  REM      DATA
602  REM  ----------
612  DATA  1,2,3,4,5,6,0,0,0,1,2,3,4
    ,5,6,3,4,5
614  DATA  1,2,3,4,5,6,7,8,9,1,2,3,4
    ,5,6,7,8,9,1,2,3,4,5,6,7,8,9
616  DATA  1,2,3,4,5,6,7,8,9,1,2,3,4
    ,5,6,7,8,9,1,2,3,4,5,6,7,8,9
618  DATA  1,2,3,4,5,6,7,8,9
619  REM  -- BOX CENTERS --
```

```
620  DATA  120,8,20,8
621  DATA  160,8,20,8
622  DATA  140,23,80,20
624  DATA  140,44,230,20
625  DATA  230,120,68,130
626  DATA  90,120,145,130
630  REM  -- WORDS ---
632  DATA  "SCORE"
634  DATA  "PUSH 'RETURN' TO START"
636  DATA  "PUSH ESC TO END"
638  DATA   "A X B   C"
640  DATA  "+","-"
642  DATA  -15,0,15,0,-15,-13,15,13
644  DATA  0,-13,0,13,15,-13,-15,13
646  DATA  212,98,212,158
650  REM  -- READ ARRAYS --
652  FOR I = 1 TO 81: READ NM(I): NEXT

656  FOR I = 1 TO 6
660  READ XO(I),YO(I),DX(I),DY(I)
662  NEXT
670  FOR I = 1 TO 6: READ WD$(I): NEXT

675  FOR I = 2 TO 5: READ X1(I),Y1(I
     ),X2(I),Y2(I): NEXT
680  FOR I = 0 TO 1: READ NX(I),NY(I
     ): NEXT
699  RETURN
700  REM  ----------
701  REM      SETUP
702  REM  ----------
704  POKE 28,255
708  CALL 62454
710 CO = 0
712  FOR I = 1 TO 4
714  GOSUB 800: GOSUB 900
```

```
716  NEXT
718 HH = YB
720  FOR I = 1 TO 21
730  HPLOT XB,YB TO XB - I,YB
732  HPLOT XE,YB TO XE + I,YB
734 YB = YB + 1: NEXT
736  FOR I = 5 TO 6
738  GOSUB 800:XH = XE
740  FOR J = 1 TO 6: GOSUB 850: NEXT

742  NEXT :YB = YB - 6:XE = XH
744  FOR J = 1 TO 6
746  HPLOT XE,YB TO XE + 35,YB
748  HPLOT XE,YB + 30 TO XE + 35,YB +
     30:YB = YB + 1: NEXT
750  REM  -- GRID --
755 XH = 25:YH = 76:XV = 37:YV = 63
760  FOR I = 1 TO 8
765  HPLOT XH,YH TO XH + 130,YH
770  HPLOT XV,YV TO XV,YV + 114
775 XV = XV + 15:YH = YH + 13: NEXT

780  HPLOT 196,120 TO 260,120
785 CO = 0:WD = 1:LX = 207:LY = 78: GOSUB
     950
790 WD = 1:LX = 207:LY = 138: GOSUB
     950
795  RETURN
800  REM  =COMPUTE END POINTS-
810  HCOLOR= CO
820 XB = XO(I) - DX(I) / 2:XE = XO(I
     ) + DX(I) / 2
830 YB = YO(I) - DY(I) / 2:YE = YO(I
     ) + DY(I) / 2
840  RETURN
850  REM     ----- HOLLOW BOX ----
```

```
860  HPLOT XB,YB TO XE,YB TO XE,YE TO
     XB,YE TO XB,YB
880 XB = XB + 1:XE = XE - 1
882 YB = YB + 1:YE = YE - 1
890  RETURN
900  REM  ------ SOLID BOX ---
910  HCOLOR= CO
920  FOR II = YB TO YE
930  HPLOT XB,II TO XE,II
940  NEXT
949  RETURN
950  REM  -- PRINT WORDS --
952  HCOLOR= CO
954  FOR II = 1 TO  LEN (WD$(WD))
956 SV =  ASC ( MID$ (WD$(WD),II,1))

958  DRAW SV - 31 AT LX,LY:LX = LX +
     10: NEXT
975  RETURN
1000  REM  -- SCORE--
1010  HCOLOR= CO
1020 SC$ =  STR$ (SC)
1030  FOR II = 1 TO  LEN (SC$):SV =
      ASC ( MID$ (SC$,II,1))
1040  DRAW SV - 31 AT NX,NY:NX = NX +
     10: NEXT
1050  RETURN
1100  REM  -- END --
1102  POKE 769,20
1105  FOR I = 1 TO 10: POKE 0,15 * I
     : CALL 768
1110  FOR J = 1 TO 10: CALL 768
1115  XDRAW  ASC ("*") - 31 AT NX(PN
     ),NY(PN) + 10: NEXT : NEXT
1120  GOSUB 410
1125  CLEAR :EN = 1: GOTO 10
```

```
1150  IF  PEEK (800) <  > 1 THEN  HGR2
     : HOME : TEXT : HTAB 18: VTAB 1
     0: PRINT "ENDED": END
1160  PRINT D$;"RUN MENU"
1200  REM  ====================
```

變數使用說明

PN ……遊戲者（搖桿）號碼。

Q …… PDL 值。

TT ……迴圈計數器。

A、B、C …… 3 個一串的閃動數字之值。

A1, A2
A3, A4 } …… (A×B±C) 的四個可能的答案。

TI ……計時器。

CO ……高解像顏色值。

NX, NY ……分數顯示之座標。

AN ……由亂數所造出的答案。

SC ……存分數之變數。

SL ……常數，超過此分數則勝利。

P1, P2 ……鍵盤及選用位址（−16336 及 −16384）。

NM$……將數字畫在畫面之暫存變數。

WD ……字串矩陣 WD$(WD) 之足碼。

D$ …… RETURN + CTRL − D。

HH, XH
YH　　……暫存用變數。
XV, YV

XB, XE
YB, YE　……方形之端點。

SV ……字串之ASCII值，減 31 後等於對應之圖形表號碼。

矩陣使用說明

PB(2) ……雙方之按鈕位址。

K(2) ……由雙方讀取之 PDL 值換算成 1 — 6 的整數。

PX(2), PY(2) ……雙方游標在字盤上的座標值。

DS(2) ……若DS(PN)=0 ，則該游標不能移動，也不能算答案，若 DS(PN)=1 且按鈕被按下，則可以移動游標或計算答案。

X1(6), Y1(6)
X2(6), Y2(6) ……游標位置之週圍位置的 6 個抵補值。

PS(2) ……格子位置（ 1 —81）。

NM(82)……存每個格子內數值的矩陣，其足碼爲 PS(PN)。

SC(2) ……雙方的分數。

WD$(6)……字串矩陣。

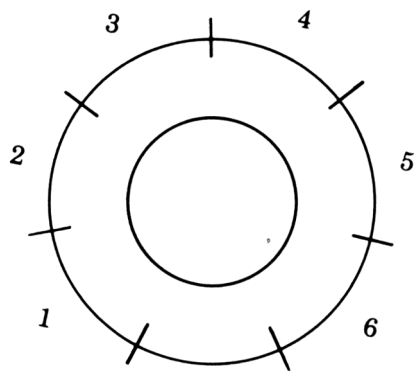
XO(6), YO(6) ……畫面上各方形的中心點座標。

DX(6),DY(6) …… 畫面上各方形的長及寬。

遊戲鍵說明

RETURN ……令遊戲開始。

ESC ……令遊戲結束。



位置 1 ：只有游標本身閃動，按下按鈕會令游標水平移動。

位置 2 ：令以游標爲中心的水平 3 個數閃動。

位置 3 ：令以游標爲中心 45°對角的 3 個數閃動。

位置 4 ：令以游標爲中心的垂直 3 個數閃動。

位置 5 ：令以游標爲中心 135° 對角的 3 個數閃動。

位置 6 ：只有游標本身閃動，按下按鈕會令游標垂直移動。

按　鈕：

當旋鈕在 1 或 6 的位置時：移動游標。

當旋鈕在 2 至 5 的位置時：計算 3 個一串閃動的數是否與答案相符？

程式說明

10—50行 ……宣告維數。

60—95行 ……至各副程式做初始設定。

100—119行 ……讀PDL(0)及PDL(1)的迴圈，倒數計時，並檢知是否按鈕被按下？

120—135行 ……將110—119行所讀到的PDL 值換成1～6的整數，如果換算值介於2～5之間，則令畫面做3個數一串的閃爍。

140—149行 ……將3個一串的數閃動之副程式。

250—269行 ……若PDL 值換算後爲1或6，則將其游標移動，若介於2～5，則將閃動的3個數代入公式（A×B±C)計算。

270—289行 ……垂直及水平移動副程式，若到達一行或一列的端點則換由下一行（或列）開始。

300—340行 ……檢查3個閃動的數是否符合公式。

345—349行 ……公式計算副程式，並且試驗3個數的各種組合情況。

350—366行 ……倒數計時，如果時限到達，則改換新答案，然後重新計時。

375—399行 ……印出分數，並且檢知何者爲勝利者。

400—439行 ……顯示初始字幕，並等候按鍵。

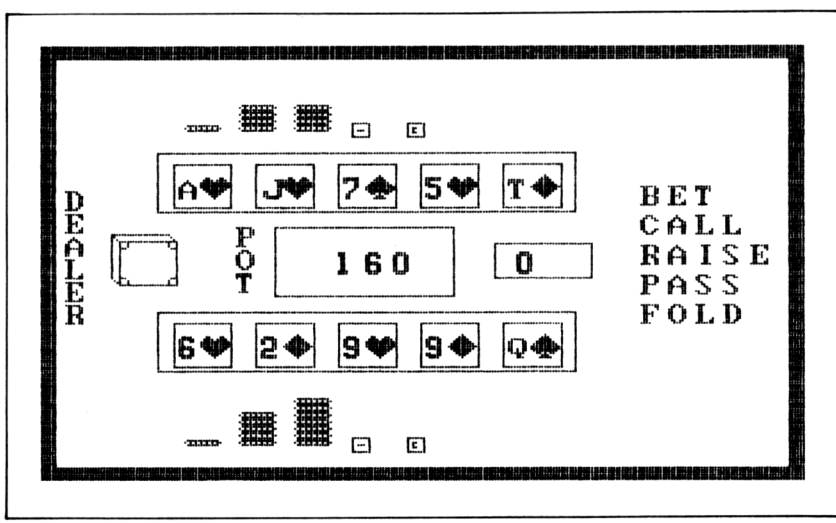
450—474行 ……以亂數方式來造出字盤矩陣之副程式。

475—495行 ……填字盤之副程式。

500—599行 ……初始變數設定，載入圖形表。
600—699行 ……讀入矩陣值。
700—795行 ……設定高解像畫面。
800—840行 ……計算方形的端點。
850—890行 ……畫方形副程式。
900—949行 ……畫實心方形。
950—975行 ……畫字形副程式。
1000—1050行……分數顯示副程式。
1100—1200行……結尾處理。

第16章

金 樸 克



遊戲說明

這是一般稱爲 PAIRS 或FULL HOUSE的樸克牌局，以遊戲的觀點來說其賭博性實大於其趣味性，我們之所以將它加以介紹，實在是因爲想參考其有趣的圖形能力，以及對於棋局之類程式的處理方式，因此其程式學習上的意義或許更大於其遊戲價值吧！

程式一開始執行，畫面上便出現各樣字幕、籌碼、牌桌等等，你必須按下 RETURN 鍵令遊戲開始，如果是按 ESC 鍵，程式便結束了。首先，發牌者（DEALER）會發給雙方一張牌，點數高的一方有權先移動游標作項目（BET、PASS、FOLD 等）選擇，如果選 BET（下注），便要再移動搖桿，使游標至適當的籌碼位置，此時按下按鈕（BUTTON），便可以將籌碼下注至牌桌上。如若要更改下注數目，可以將游標移至－，按下按鈕，可以將下注收回，再重新押注，當押注金額決定無誤之後，還要移動游標至 POT 位置內，按下按鈕，目前的押注金才會加入到總額內，接著便由對方下注。雙方下注完畢後，移動游標至 CALL，按下按鈕，便可以叫下一張牌，當五張牌全部發完後，勝負就分曉了。

遊戲的規定是每次的押注金要與先押注者的金額相等，不過可以在雙方均下注完畢後，再按下 RAISE 便可以增加金額，但最多不可超過 3 次。如果認爲不可能得勝，則可以按下 FOLD 認輸此局。按下 C 可以兌換成較小的籌碼，PASS 則是放棄此番的押注權。

程式

```
]LIST

10  REM ****************************
20  REM ** POKER! POKER! POKER! **
30  REM ****************************
```

```
38  DIM CD$(53),CA$(15),C$(15)
40  DIM SV(5),CV(15)
44  DIM MX(1,7),MY(1,7)
50  DIM CX(1,7),CY(1,7)
54  DIM R(5),H(5)
58  DIM PY(6),BV(4)
60  REM ====POKE SOUND====
61  FOR I = 0 TO 22: READ WW
62  POKE 768 + I,WW: NEXT I
63  DATA 169,10,133,1,173,48
64  DATA  192,136,208,4,198,1
65  DATA  240,8,202,208,246
66  DATA  166,0,76,4,3,96
70  REM  --------------
75  REM        MAIN
80  REM  --------------
90  GOSUB 600: GOSUB 700
92  GOSUB 750: GOSUB 875
94  GOSUB 900: GOSUB 400
96 PN = 0: GOTO 100
100  REM  -- DEAL HAND ---
104 JJ = 0
108  FOR G = 1 TO 5
110  FOR I = 1 TO 2:JJ = JJ + 1: GOSUB
     925: GOSUB 175:PN = 1 - PN: NEXT
116  IF G = 5 THEN 124
120  GOSUB 200:KX = KX + 30
124 PN = 0: NEXT
125  REM   --- HAND OVER ---
126  POKE P1,0
128  IF TI = 1 THEN  GOTO 134
130 PN = WN: GOSUB 575
131  IF B2 = 2 THEN  GOTO 1100
132  IF BR = 1 AND BP <  > WN THEN  GOTO
     1100
133 BR = 0:SW$ = "":B2 = 0
```

```
134  IF  PEEK (P2) = 155 THEN  GOTO 115
     0
136  IF  PEEK (P2) <  > 141 THEN 134
138  HCOLOR= 0: FOR I = 1 TO 21
140  HPLOT 48,51 + I TO 192,51 + I: HPLOT
     48,120 + I TO 192,120 + I
142  NEXT : GOSUB 150
146  GOTO 94
150  REM  -- RESET NEW HAND --
155 KX = 6:CO = 0:BH = 0:PS = 0:RL = 0
156  POKE 0,5: POKE 769,10
160  IF TI = 1 THEN TI = 0: GOTO 170
165 BT = PT:LX = 110:LY = 100: GOSUB 10
     00:PT = 0
170 BT = 0
174  FOR I = 1 TO 5:HC(0,I) = 0:HC(1,I)
      = 0: NEXT : RETURN
175  REM  -- SHOW CARD --
176  POKE 232,168: POKE 233,97
178  HCOLOR= 3
180 HX = SX(PN) + KX + 2:HY = SY(PN) -
     KY - 4
182 CR =  VAL (C$(JJ))
184  DRAW  ASC (CA$(CR)) - 31 AT HX,HY
188  POKE 232,120: POKE 233,105
190 HX = HX + 8
192  DRAW  VAL (S$(JJ)) + 1 AT HX,HY
195  CALL 768: RETURN
200  REM  -- CHOOSE PLAY --
202  IF BR = 1 THEN  RETURN
205 KK = 1
210 Q =  PDL (PN)
215 II =  INT (Q / 55) + 1
220  HCOLOR= 0: DRAW 7 AT PX,PY(KK): XDRAW
     7 AT PX,PY(II):KK = II
225  IF  PEEK (PB(PN)) < 128 THEN  GOTO 210
```

```
230  IF  PEEK (PB(PN)) > 127 THEN 230
235  XDRAW 7 AT PX,PY(KK)
240  ON KK GOSUB 250,350,360,370,375
241  IF BR = 1 THEN  RETURN
242  IF SW$ = "F" THEN  GOTO 249
243  IF SW$ = "P" THEN  GOTO 249
244  IF SW$ = "R" THEN SW$ = "": GOTO 2
     40
246  IF SW$ = "C" THEN  GOTO 249
248  GOTO 205
249 SW$ = "": RETURN
250  REM  --- BET --
252  IF SW$ = "B" THEN  RETURN
256 KK = 1:SW$ = "B":PA = 0
258 Q =  PDL (PN)
260 II =  INT (Q / 40) + 1: GOSUB 290
264  IF  PEEK (PB(PN)) < 128 THEN  GOTO
     258
266  IF  PEEK (PB(PN)) > 127 THEN 266
268  XDRAW 7 AT MX(PN,KK),MY(PN,KK)
270  ON KK GOSUB 390,390,390,300,320,34
     0,375
274  IF SW$ = "F" THEN  RETURN
276  IF PS > 1 THEN PS = 1: RETURN
280  GOTO 258
290  REM  -- DRAW CURSOR BAL --
292  HCOLOR= 0: DRAW 7 AT MX(PN,KK),MY(
     PN,KK): XDRAW 7 AT MX(PN,II),MY(PN
     ,II)
294 KK = II: RETURN
300  REM  -- CLEAR BET --
302 BB = PT:PT = BT: GOSUB 577:PT = BB
305 CO = 0:LX = 175:LY = 100: GOSUB 100
     0:BT = 0:CO = 3:LX = 175: GOSUB 10
     00
310  RETURN
```

```
320  REM  -- CHANGE --
322 CH$ = "CH"
324  IF CN(PN,2) = 0 AND CN(PN,3) > 0 THEN
     KK = 3: GOSUB 390: FOR I = 1 TO 5:
     KK = 2: GOSUB 380: NEXT
330  IF CN(PN,1) < 4 AND CN(PN,2) > 0 THEN
     KK = 2: GOSUB 390: FOR I = 1 TO 2:
     KK = 1: GOSUB 380: NEXT
338 CH$ = "": RETURN
340  REM  --ADD TO POT --
342 PS = PS + 1
344  IF BR = 1 AND PN = BP THEN  GOTO 3
     46
345  IF PS > 1 AND BH <  > BT THEN PS =
     1: RETURN
346 BH = BT:BT = PT:CO = 0:LX = 110:LY =
     100: GOSUB 1000
347 PT = BH + PT:BT = PT:CO = 3:LX = 11
     0: GOSUB 1000
348  IF PS = 1 THEN PN = 1 - PN
349 BT = BH: GOSUB 305: RETURN
350  REM -- CALL --
352  IF PS = 0 THEN  RETURN
353 RL = 0:PN = 1 - PN
354 SW$ = "C"
355 PS = 0: RETURN
360  REM  -- RAISE --
362  IF PS = 0 THEN  RETURN
364  IF RL = 3 THEN  RETURN
365 SW$ = "R":PS = 0
366  GOSUB 250:RL = RL + 1
367 KK = 1
368  RETURN
370  REM  -- PASS --
372  IF SW$ = "B" THEN  RETURN
373 PA = PA + 1: IF PA = 2 THEN PA = 0:
```

```
     SW$ = "P":PN = 1 - PN: RETURN
374 PN = 1 - PN: RETURN
375  REM  -- FOLD HAND --
376  IF BT <  > 0 THEN  RETURN
377 SW$ = "F":WN = 1 - PN:G = 5
378  GOSUB 305
379  RETURN
380  REM  --- DRAW CHIP ---
381  HCOLOR= 3:CN(PN,KK) = CN(PN,KK) +
     1
382  DRAW 6 AT CX(PN,KK),CY(PN,KK) - CW
      * CN(PN,KK)
384  CALL 768
386  RETURN
389  REM  --- ERASE CHIP ---
390  IF BT + BV(KK) > 50 AND CH$ <  > "
     CH" THEN  RETURN
391  HCOLOR= 0: IF CN(PN,KK) = 0 THEN  RETURN

392  XDRAW 6 AT CX(PN,KK),CY(PN,KK) - C
     W * CN(PN,KK):CN(PN,KK) = CN(PN,KK
     ) - 1
393  IF CH$ = "CH" THEN  RETURN
394  CALL 768: GOSUB 395: RETURN
395  REM  --- BET HOLD --
396 CO = 0:LX = 175:LY = 100: GOSUB 100
     0:BT = BV(KK) + BT
397 CO = 3:LX = 175:LY = 100: GOSUB 100
     0
398  IF CN(PN,1) = 0 AND CN(PN,2) = 0 AND
     CN(PN,3) = 0 THEN BR = 1:BP = PN:B
     2 = B2 + 1
399  RETURN
400  REM  --- COMPUTE RANK --
401 PN = 0: FOR I = 1 TO 9 STEP 2: GOSUB
     455: NEXT
```

```
402  GOSUB 465: GOSUB 475: GOSUB 416: GOSUB
     415
404 PN = 1: FOR I = 2 TO 10 STEP 2: GOSUB
     455: NEXT
406  GOSUB 465: GOSUB 475: GOSUB 416: GOSUB
     415
408  IF RK(0) = RK(1) THEN I = 0: GOSUB
     420
410  IF RK(0) > RK(1) THEN WN = 0
412  IF RK(0) < RK(1) THEN WN = 1
414  RETURN
415  FOR I = 1 TO 5:SV(I) = 0:R(I) = 0:
      NEXT : FOR I = 1 TO 14:CV(I) = 0:
      NEXT : RETURN
416 J = 0: ON RK(PN) GOSUB 450,425,430,
     435,450,450,440,445,450
418  RETURN
419  REM  -- TIE HAND--
420 WN = 1:I = I + 1: IF I = 6 THEN TI =
     1: RETURN
421  IF HC(0,I) = HC(1,I) THEN 420
422  IF HC(0,I) > HC(1,I) THEN WN = 0
423  RETURN
424  REM  -HI CARD ORDERING -
425  FOR I = 1 TO 4
426  IF R(I) = 2 THEN HC(PN,1) = H(I):J
      = J + 1
427  IF R(I) = 1 THEN HC(PN,5 - I + J) =
     H(I)
428  NEXT : RETURN
429  REM  ------------
430  FOR I = 1 TO 3
431  IF R(I) = 1 THEN HC(PN,3) = H(I):J
      = J + 1
432  IF R(I) = 2 THEN HC(PN,4 - I + J) =
     H(I)
```

```
433  NEXT : RETURN
434  REM ------------
435  FOR I = 1 TO 3
436  IF R(I) = 3 THEN HC(PN,1) = H(I):J
      = J + 1
437  IF R(I) = 1 THEN HC(PN,4 - I + J) =
     H(I)
438  NEXT : RETURN
439  REM ------------
440  FOR I = 1 TO 2
441  IF R(I) = 3 THEN HC(PN,1) = H(I)
442  IF R(I) = 2 THEN HC(PN,2) = H(I)
443  NEXT : RETURN
444  REM  ------------
445  FOR I = 1 TO 2
446  IF R(I) = 4 THEN HC(PN,1) = H(I)
447  IF R(I) = 1 THEN HC(PN,2) = H(I)
448  NEXT : RETURN
449  REM  ----------------
450  FOR I = 1 TO 5:HC(PN,6 - I) = H(I)
     : NEXT : RETURN
454  REM  ----------------
455  REM  --- STORE CARD VALUE ---
456 S$(I) =  MID$ (CD$(I),1,1)
458 C$(I) =  MID$ (CD$(I),2,2)
460 J =  VAL (S$(I)):SV(J) = SV(J) + 1
462 J =  VAL (C$(I)):CV(J) = CV(J) + 1
463  RETURN
465  REM  --- STORE HAND ---
466 J = 1:CC = 0
467  FOR I = 2 TO 14: IF CV(I) = 0 THEN
      GOTO 472
468 R(J) = CV(I):H(J) = I:CC = CC + CV(
     I)
469  IF CC = 5 THEN I = 14
470 J = J + 1
```

```
472  NEXT
474  RETURN
475  REM  *** DETERMINE RANK ***
476 RK(PN) = 1
477  IF R(5) > 0 THEN  GOSUB 500: GOTO
     485
478  IF R(4) > 0 THEN  GOSUB 520: GOTO
     485
480  IF R(3) > 0 THEN  GOSUB 540: GOTO
     485
482  IF R(2) = 1 OR R(2) = 4 THEN  GOSUB
     550: GOTO 485
484  GOSUB 560
485  RETURN
499  RETURN
500  REM  --- STRAIGHT ---
502 CN = H(1):RK(PN) = 1
504  IF CN > 10 THEN  GOSUB 515: GOTO 5
     10
506  FOR I = 1 TO 5:SC = CV(CN) + SC:CN
      = CN + 1: NEXT
507  GOSUB 515
508  IF SC = 5 AND RK(PN) = 6 THEN RK(P
     N) = 9: GOTO 510
509  IF SC = 5 THEN RK(PN) = 5
510 SC = 0: RETURN
515  REM -- FLUSH -----
516 RK(PN) = 1
517  FOR I = 1 TO 4: IF SV(I) = 5 THEN
     RK(PN) = 6
518  NEXT : RETURN
520  REM  -- PAIR --
522 RK(PN) = 2
530  RETURN
540  REM  - 2-PAIR/3-KIND -
542 RK(PN) = 3
```

```
544  FOR I = 1 TO 3
546  IF R(I) = 3 THEN RK(PN) = 4:I = 3
548  NEXT
549  RETURN
550  REM  --- 4-KIND---
552 RK(PN) = 8: RETURN
560  REM  ------ FULL HOUSE ---
565 RK(PN) = 7: RETURN
575  REM  ------CONVERT TO CHIPS ----
576  FOR I = 1 TO 10: POKE 0,10 * I: POKE
     769,10 * I: CALL 768: NEXT : POKE
     0,5: POKE 769,10
577 PH = PT: HCOLOR= 3
578  FOR L = 3 TO 1 STEP  - 1:T1 =  INT
     (PT / CC(L))
580  IF PT = 0 OR T1 = 0 THEN  GOTO 584

582  GOSUB 590
584  NEXT : GOTO 599
590 T2 = T1: FOR J = 1 TO T1
592  IF CN(PN,L) > 9 THEN T2 = J - 1:J =
     T1: GOTO 596
594 CN(PN,L) = CN(PN,L) + 1: CALL 768: DRAW
     6 AT CX(PN,L),CY(PN,L) - CW * CN(P
     N,L)
596  NEXT
598 PT = PT - (CC(L) * T2): RETURN
599 PT = PH: RETURN
600  REM  ------------
601  REM     INITIALIZE
602  REM  ----------------
605 BV(1) = 5:BV(2) = 10
606 BV(3) = 50:CW = 3
607 CC(1) = 5:CC(2) = 10
608 CC(3) = 50:BR = 0:BB = 0
610 I = 1:KY = 2:KX = 6:PX = 210
```

```
611 P1 =  - 16368:P2 =  - 16384
612 PB(0) =  - 16287:PB(1) =  - 16286
615 D$ =  CHR$ (13) +  CHR$ (4)
620  IF EN = 1 THEN  GOTO 645
625  PRINT D$;"BLOAD POKER-TBL"
630  PRINT D$;"BLOAD ALPHA-TBL"
640  IF  PEEK (800) <  > 1 THEN R =  RND
     ( -  PEEK (78) -  PEEK (79) * 256)

645  HGR2 : SCALE= 1
646  ROT= 0: HCOLOR= 3
650  POKE P1,0
655  POKE 0,5: POKE 769,10
660  POKE 232,120: POKE 233,105
665  DEF  FN RD(RR) =  INT ( RND (1) *
     RR)
670  FOR J = 1 TO 4: FOR K = 2 TO 14
675 CD$(I) =  STR$ (J) +  STR$ (K):I =
     I + 1
680  NEXT : NEXT
685  FOR I = 0 TO 1
690  FOR J = 1 TO 3
695 CN(I,J) = 5: NEXT : NEXT
699  RETURN
700  REM  --- DATA ----
704  DATA  60,45,80,45,100,45,120
706  DATA  45,140,45,100,100,210,120
708  DATA  60,183,80,183,100,183,120,18
     3,140,183,100,100,210,120
710  REM  --- FACE VALUE --
712  DATA  "2","3","4","5","6","7"
714  DATA  "8,","9","T","J","Q","K","A"

715  REM  ---- COORD DATA ----
720  DATA  70,83,96,109,122
724  DATA  278,190,152,26
```

```
726  DATA  65,30,35,15
730  REM  --- WORDS ----
732  DATA  "BET","CALL","RAISE","PASS",
     "FOLD","DEALER","POT"
734  DATA  "RETURN*TO*PLAY","ESC*TO*END
     ","CHAMP"
735  REM  --- READ ARRYS ---
736  FOR I = 0 TO 1: FOR J = 1 TO 7: READ
     MX(I,J),MY(I,J): IF J < 6 THEN CX(
     I,J) = MX(I,J) - 5:CY(I,J) = MY(I,
     J) - 5
738  NEXT : NEXT
740  FOR I = 2 TO 14: READ CA$(I): NEXT

742  FOR I = 1 TO 5: READ PY(I): NEXT
744  FOR I = 1 TO 4: READ DX(I),DY(I): NEXT
746  FOR I = 1 TO 10: READ WD$(I): NEXT

749  RETURN
750  REM  ---------------
751  REM      SET UP
752  REM  ---------------
756 II = 1:CO = 3
758 XO = 140:YO = 95: GOSUB 800: GOSUB
     850
759  DRAW 10 AT 30,105
760 XO = 120:YO = 60:II = 2: GOSUB 800:
     SX(0) = XB:SY(0) = YE
762 XO = 120:YO = 130: GOSUB 800:SX(1) =
     XB:SY(1) = YE
763 II = 3:XO = 120:YO = 95: GOSUB 800
764 II = 4:XO = 185:YO = 95: GOSUB 800
765 LX = 10:LY = 72:YL = 10:XL = 0:WD =
     6: GOSUB 950
766 LX = 73:LY = 88:YL = 10:XL = 0:WD =
     7: GOSUB 950
```

```
768 LX = 220:LY = 70:YL = 0:XL = 10
770  FOR WD = 1 TO 5: GOSUB 950:LX = 22
     0:LY = LY + 13: NEXT
773 XB = 30:YB = 105
775  REM  ----- SET CHIP ---
778  FOR PN = 0 TO 1: FOR J = 1 TO 3: FOR
     I = 1 TO 5
780  DRAW 6 AT CX(PN,J),CY(PN,J) - CW *
     I
788  NEXT : NEXT : NEXT
790  FOR PN = 0 TO 1: FOR J = 4 TO 5
792  DRAW 4 + J AT CX(PN,J),CY(PN,J)
795  NEXT : NEXT :PN = 0: RETURN
800  REM  ----- DRAW SQUARE ---
810  HCOLOR= CO
820 XB = XO - DX(II) / 2:XE = XO + DX(I
     I) / 2
830 YB = YO - DY(II) / 2:YE = YO + DY(I
     I) / 2
840  HPLOT XB,YB TO XE,YB TO XE,YE TO X
     B,YE TO XB,YB
849  RETURN
850  FOR I = 1 TO 5
853  CALL 768
854  HPLOT XB,YB + I TO XE,YB + I: HPLOT
     XB + I,YB TO XB + I,YE
855  CALL 768
856  HPLOT XB,YE - I TO XE,YE - I: HPLOT
     XE - I,YB TO XE - I,YE
860  NEXT
870  RETURN
875  REM  --- START PLAY -----
876 CO = 3:WD = 8:LX = 50:LY = 65: GOSUB
     950
877 WD = 9:LX = 50:LY = 135: GOSUB 950
878 ZZ =  PEEK (P2)
```

```
879  IF ZZ = 155 THEN  GOTO 1150
880  IF ZZ = 141 THEN  GOTO 886
882  GOTO 878
886 CO = 0:WD = 8:LX = 50:LY = 65: GOSUB
     950
888 WD = 9:LX = 50:LY = 135: GOSUB 950
889  RETURN
900  REM  --- SHUFFLE ---
902  FOR I = 52 TO 2 STEP  - 1
904 J =  FN RD(I) + 1:T$ = CD$(I)
908 CD$(I) = CD$(J):CD$(J) = T$
910  NEXT
912  FOR II = 1 TO 2
914  FOR I = 1 TO 10 STEP 2
916  XDRAW 1 AT 30 - I,106 - I: CALL 76
     8: NEXT
918  FOR I = 10 TO 1 STEP  - 2
920  XDRAW 1 AT 30 - I,106 - I: CALL 76
     8: NEXT
922  NEXT
924  RETURN
925  REM  --- DEAL ---
926 X = XB:Y = YB
930 A = SX(PN) + KX - XB:B = SY(PN) - K
     Y - YB
932 SQ =  SQR (A * A + B * B)
934 UX = A / SQ:UY = B / SQ
936  XDRAW 1 AT X,Y
938  FOR K = 0 TO SQ STEP 5
940  XDRAW 1 AT X,Y
942 X = XB + UX * K:Y = YB + UY * K
944  XDRAW 1 AT X,Y: NEXT
945 K = SQ: XDRAW 1 AT X,Y:X = XB + UX *
     K:Y = YB + UY * K: XDRAW 1 AT X,Y:
      RETURN
950  REM  ----- PRINT WORDS ---
```

```
952  POKE 232,168: POKE 233,97: HCOLOR=
     CO
954  FOR II = 1 TO  LEN (WD$(WD))
956 SV =  ASC ( MID$ (WD$(WD),II,1))
958  DRAW SV - 31 AT LX,LY:LX = LX + XL
     :LY = LY + YL: NEXT
960  POKE 232,120: POKE 233,105
975  RETURN
1000  REM  ----- PRINT BETS ---
1010  POKE 232,168: POKE 233,97: HCOLOR=
     CO
1020 BT$ =  STR$ (BT)
1030  FOR II = 1 TO  LEN (BT$):SV =  ASC
     ( MID$ (BT$,II,1))
1040  DRAW SV - 31 AT LX,LY:LX = LX + 1
     0: NEXT
1045  POKE 232,120: POKE 233,105
1050  RETURN
1100  REM  --- END ---
1105  IF WN = 0 THEN LY = 55
1110  IF WN = 1 THEN LY = 140
1120  FOR I = 1 TO 10
1125  POKE 0,30: POKE 769,50:CO = 0:WD =
     10:LX = 210: GOSUB 950: CALL 768
1130  POKE 0,200: POKE 769,50:CO = 3:WD
      = 10:LX = 210: GOSUB 950: CALL 76
     8: NEXT
1132  POKE P1,0
1139  REM  --- LOOP FOR INPUT ----
1140  IF  PEEK (P2) = 141 THEN  CLEAR :
     EN = 1: GOTO 10
1145  IF  PEEK (P2) = 155 THEN  GOTO 11
     50
1148  GOTO 1140
1149  REM  ---END /LINK ----
1150  IF  PEEK (800) <  > 1 THEN  HGR2
```

```
        : TEXT : HOME : END
1160   PRINT D$;"RUN MENU"
1200   REM  ==============
```

圖號	1	2	3	4	5	6	7
圖案							

圖號	8	9	10
圖案			

圖案及其圖形號碼

```
*6978.6AF3

6978- 0A 00 16 00 3F 00 69 00
6980- 90 00 B8 00 E3 00 F8 00
6988- 00 01 10 01 21 01 2D 2D
6990- 2D 2D 2D 2D 2D 2D 2D 2D
6998- 25 24 24 24 24 24 24 24
69A0- 24 24 3F 3F 3F 3F 3F 3F
69A8- 3F 3F 3F 3F 37 36 36 36
69B0- 36 36 36 36 36 36 00 49
69B8- 09 05 38 3F 28 2D 2D 38
69C0- 3F 3F 3F 28 2D 2D 2D 2D
69C8- 38 3F 3F 3F 3F 27 2D 2D
69D0- 2D 2D 2D 3C 3F 3F 3F 3F
69D8- 67 2D 6D 2D E5 FF 1B 3F
69E0- 00 49 09 05 38 3F 28 2D
69E8- 2D 38 3F 3F 3F 28 2D 2D
```

```
69F0- 2D 2D 38 3F 3F 3F 3F 67
69F8- 2D 2D 2D 2D 1C 3F 3F 3F
6A00- 0C 2D 2D 1C 3F 0C 05 00
6A08- 49 49 39 3F 67 2D 1C FC
6A10- 3F 07 28 2D 2D 2D 2E 2C
6A18- 2E 20 3F 3F 3F 3F 3F 0C
6A20- 2D 2D 2D 2D DC 3B 3F 28
6A28- 2D 25 3F 3F 0C 2D 05 00
6A30- 49 49 39 3F 67 2D 1C 1F
6A38- 3F 28 2D 2D 35 25 2D 38
6A40- 3F 3F 3F 3F 27 2D 2D 2D
6A48- 2D 2D 1C 3F 3F 3F 3F 0C
6A50- 2D 2D 2D 1C 3F 3F 0C 2D
6A58- 1C 07 00 2D 2D 2D 2D 2D
6A60- 2D E0 3F 3F 3F 3F 3F BF
6A68- 0D 0D 0D 0D 0D 0D 05 00
6A70- 2D 38 3F 2C 2D 1C 3F 00
6A78- 24 24 24 2D 2D 2D 36 36
6A80- 36 3F 3F 27 40 2D 05 00
6A88- 24 24 24 2D 2D 2D 36 36
6A90- 36 3F 3F 67 20 2C 16 07
6A98- 00 2D 2D 2D 2D 2D 2D 2D
6AA0- 2D 2D 2D 25 24 24 24 24
6AA8- 24 24 24 24 24 3F 3F 3F
6AB0- 3F 3F 3F 3F 3F 3F 3F 37
6AB8- 36 36 36 36 36 36 36 36
6AC0- 3E 1C 1C 24 24 24 24 24
6AC8- 24 24 24 24 74 0E C4 2D
6AD0- 2D 2D 2D 2D 2D 2D 2D 2D
6AD8- 2D 0E 0E 96 3A 07 20 DF
6AE0- DB DB DB DB F6 B7 92 92
6AE8- 92 12 AD 6E 49 49 49 49
6AF0- 21 0C 2D 00
```

圖案圖形表

變數使用說明

PN ……遊戲者（即搖桿）號碼

I，G……迴路變數。

JJ ……發牌及洗牌時，用以顯示樸克牌之矩陣足碼變數。

BR ……中斷旗號，當遊戲者之任何一人輸光全部籌碼時，BR=1，否則BR=0。

BP ……記錄輸光籌碼者的號碼（0號或1號）。

B2 ……記錄輸光籌碼者的人數（1或2個）。

KX ……樸克牌在桌面上的水平座標距離。

KY ……樸克牌在桌面上的垂直座標距離。（即與對方的距離）。

P1，P2 ……按鍵及按鍵選用（ strobe ）位址（—16384及—16368）。

TI ……平手旗號，如果雙方牌點平分，則TI=1。

WN ……贏者號碼（0或1號）。

CO ……高解像顏色號碼。

LX，LY ……字形顯示之座標位置。

BT ……該次押注金額。

BT$……記錄押注金額的字串。

PT ……押注的總額。

BB，BH ……暫存 BT 值的變數。

HX，HY ……畫樸克牌點數的水平及垂直座標。

CR ……樸克牌數值（當矩陣足碼用）。

Q……搖桿值。

II ……由Q值轉換成1～6的整數值，搖桿以此來控制畫面游標位置。

KK ……暫存 II 的值。

PX ……畫面游標的水平位置。

SW$……游標所選到的項目代碼（ bet=B, call=C, fold=F, raise=R, pass=P）。

PA ……Pass 的計數，如果雙方均Pass ，則發下一對牌。

PS ……遊戲者旗號（如果 PS=1 ，則讀另一人之搖桿及按鈕）。

CH$……變換至對方的旗號。

RL ……增加賭金之限制，最多不可增加超過 3 次。

J ……泛用計數器。

CW ……記錄籌碼的高度。

CC ……於 465 — 474 行當中，用於記錄每手牌時之計數。

CN ……記錄第一張樸克牌的號碼，用以檢查是否得到 STRAIGHT 。

SC ……STRAIGHT 計數器，若 SC=5 ，則該手牌爲 STRAIGHT 。

T1 ……換籌碼之暫存變數。

T2 ……換籌碼之暫存變數。

D$ …… RETURN + CTRL — D 。

XO, YO ……畫面設定時，各方形之中央座標。

XL, YL ……字形顯示副程式之字形垂直及水平座標間隔。

XB, XE, YB, YE …方形圖案之頂點（四角）座標。

WD ……字形字串矩陣之足碼。

X,Y……發牌時之樸克牌座標。

A,B……發牌者將樸克牌發至桌面之垂直及水平距離。

SQ ……由A，B垂直、水平分量構成之向量。

UX,UY ……垂直及水平方向的單位向量。

SV ……字形之ASCII碼，減 31 之後就成爲該字形的圖形號碼。

矩陣使用說明

PY(7) ……項目（Bet,Call，……）選擇之垂直座標。

PB(1) ……按鈕位址（BUTTON(0)及BUTTON（1））。

MX(1,7),MY(1,7)…畫面游標在押注時之座標位置，第一個足碼爲遊戲者號碼0～1，第二個足碼爲搖桿值0～7。

CN(1,3) ……每一疊籌碼之單位值（5,10,50)。

BV(3) ……每一個籌碼押注值，BV(1)=5,BV(2)=10,BV(3)=50。

CX(1,7),CY(1,7)…每疊籌碼位置的基準位置座標。

S$(4) ……花色號碼（1＝紅心，2＝鑽石，3＝梅花，4＝黑桃）。

C$(15)……牌面值（2～14,11=Jack,12=Queen

，13=King，14=Ace）。

CD$(53)……全副 52 張牌之花色及牌面值之對應碼，譬如CD$(5)="15"= 紅心5 。

CA$(15)……代表牌面字形之字串，（ 11=Jack → "J"）。

CV(15)……用牌面值之張數，譬如CV(3)=2 ，表示有兩張 3 ，CV(14)=3表示有3 張A 。

SU(5) ……已發出去之該花色牌總數，譬如9V(4)=7，表示黑桃已發出 7 張了。

H(5) ……一手牌當中，每張牌的面值。

HC(1,5) ……將 5 張牌以大小順序排列起來，以便當雙方的牌級相同時，用以比較分數高低。

PK(1) ……遊戲者的牌級對應的值。

1 = no pair	6 = slush
2 = pair	7 = full house
3 = two pair	8 = 4 − kind
4 = 3 − kind	9 = royal flush
5 = straight	

CC(3) ……當一手牌結束時，押注歸入贏家時，用以計算每疊籌碼之數目。

R(5)……牌局矩陣。

DX(4),DY(4) ……要畫在螢幕上的每個方形的長寬值。

WD$(10)……字形字串矩陣。

遊戲鍵說明

RETURN ……開始牌局。

ESC ……結束牌局。

搖桿（0或1）……在項目選擇及押注狀態下，移動游標來選所欲的項目，按下按鈕（0或1）便可選到此時游標所在的項目。

項目選擇：

Bet…… 押注。

Call…… 叫牌。

Raise…… 增加金額。

Pass …… Pass

Fold …… 認輸此局。

押注選擇：

左邊籌碼…… 5

中間籌碼…… 10

右邊籌碼…… 50

— …… 更改下注金額。

C …… 換籌碼。

Pot…… 若移動游標至Pot方格內，按下按鈕，便可將押注金額加入至總額內。

Fold …… 認輸此局。

程式說明

10—58行 ……宣告維數。

60—96行 ……GOSUB 做起始設定。

100—124行 ……牌局迴路，由發牌開始，分支到押注迴路，如果5張牌皆發完，則至一局結束副程式(Hand over)。

125—146行 ……一局結束(Hand over)副程式，重置鍵盤選用(−16368) 位址。檢查是否平手，若是，則等候下一手牌，若否，則將賭金搬至贏者的籌碼上。若有人已輸光籌碼，則結束牌局，若否，則重置變數，重新一手牌。

150—174行 ……重置變數，開始新牌局。

175—195行 ……畫出所發牌的點數及花色。

200—235行 ……在項目(Bet Call Raise ……)選擇副程式，讀取PDL，移動游標至所選項目的位置。

240—249行 ……依前面所選之項目至該項目所對應的副程式。

250—280行 ……押注選擇副程式，讀取PDL(0或1)值，至對應的籌碼位置，如果二者均已下注(遊戲者旗號(PS)＞1)，則回到項選擇副程式。

290—294行 ……在舊位置消去游標，然後在新位置畫出新游標。

300—310行 ……清除目前下注金額，然後重新下注。

320—338行 ……換籌碼副程式。若某一籌碼已用光則以較高面值兌換之。

340—349行 ……將押注加入到賭金總額，若押注金不等於先前之押注金額，則不能押注總額（遊戲規則）。

350—355行 ……叫牌，換手。

360—368行 ……增加押注金額，若目前還沒有押注金（第一次），則RETURN，若增加金額已達3次，也RETURN。若符合增加金額條件，則經由250行的Bet副程式接受增加的金額。

370—374行 ……Pass 副程式，若已有一方下注（bet 被選上），則不能 Pass 。若雙方均Pass，則設定SW$=〝P〞。

375—379行 ……認輸此局副程式，設定對方勝利，結束發牌，重新一局。

380—386行 ……畫籌碼副程式，改變籌碼計數，在籌碼堆上畫出籌碼。

390—394行 ……消去籌碼副程式，若押注值大於50 ，則不消去籌碼，若否，則由籌碼堆上消去籌碼，並加入至押注上。

395—399行 ……加入押注副程式，若籌碼已用完，則設定

中斷旗號 (BR=1)。

400—415行 ……計算該手牌的牌級，重置各矩陣變數。

416—418行 ……至各對應的牌面點數處理副程式。

420—423行 ……平手副程式。

424—454行 ……依牌級將牌依大小次序排列。

455—463行 ……將CD$(I)分解成花色及牌面點數，將相同點數牌的張數存入矩陣CV(J)。

465—474行 ……將CV(J) 的值存入R(J)。

475—485行 ……檢查〝R〞矩陣，並至對應的副程式內。

500—510行 ……檢查牌局是否爲Straight (SC=5)或Straight flush (SC=5且SV(I)=5)，若否，則牌級爲1。

515—518行 ……檢查是否flush (SV(I)=5)。

520—530行 ……Pair 級。

540—549行 ……Two Pair或3—Kind(R(I)=3)級。

550—552行 ……4張同點(4—Kind)級。

560—565行 ……Full house 級。

575—599行 ……將Pot方格內的賭金，分解成5,10,50，分別放入贏方的籌碼堆裡。

600—699行 ……初始設定，載入圖形表。若是磁帶用者，請依前述方式修改。

700—749行 ……讀入矩陣資料。

750—795行 ……畫面設定。

800—849行 ……計算四角座標，然後畫出方形。

850—870行 ……畫圖框。

875—889行 ……字形顯示，等候按鍵。

900—924行 ……洗牌副程式。

925—945行 ……計算發牌之相對距離，定出垂直、水平之單位向量，沿所計算的路線畫出紙牌（發牌動作）。

950—975行 ……字形顯示副程式。

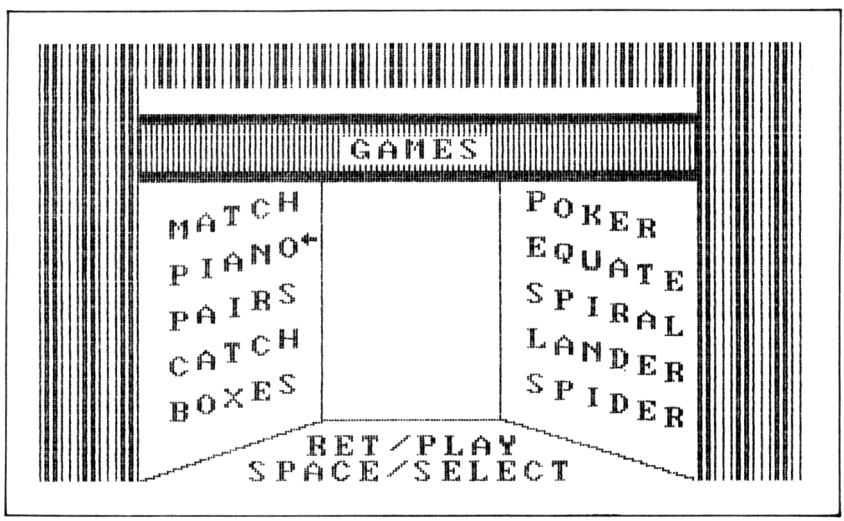
1000—1050行……押注顯示副程式。

1100—1148行……結束副程式，發聲，顯示贏家，等候開始新牌局，若開始新牌局，則設定EN=1，若否，則結束遊戲。

1150—1200行……結束程式，若由menu 來，則回menu。

第17章

節 目 單



使用說明

這個程式，本身不是一個遊戲，但却是供前述各個遊戲程式使用的一個公用程式。節目單這個程式的主要目的，是便利你去選擇前面你所鍵入並存入磁片的各種遊戲，它也提供了您在高解析圖形中以兩頁（ two page）交互顯示圖形的例子，要選擇任何遊戲時，只要按 space 鍵控制指標到你所要的遊戲名稱前面，然

後按下RETURN 程式就會自動的將您所要的遊戲載入，假如在你的磁碟之中並沒有這項遊戲時，畫面也會將錯誤訊息顯示在螢幕上。如果您使用的是卡式磁帶，請略去本章。

程式

```
]LIST

10  REM  ****************************
20  REM  **    MENU  PROGRAM!!!    **
30  REM  ****************************
40  REM
45  REM
50  DIM WD$(16)
60  REM  --------------
70  REM     MAIN
80  REM  --------------
85  REM
90  GOSUB 400: GOSUB 500
95  GOSUB 600: GOTO 100
100  REM  -----------
101  REM  READ KEY
102  REM  -----------
105 Z =  PEEK (P2)
115  IF Z = 160 THEN  GOSUB 180: GOTO 135
120  IF Z = 141 THEN  GOTO 800
125  IF Z = 155 THEN  FOR I = 1 TO 15: HGR2
     : TEXT : HOME : HTAB 8: VTAB 12: PRINT
     WD$(13): NEXT : END
130  GOTO 105
135  POKE P1,0: GOTO 105
180  REM  --MOVE SELECTER --
```

```
182  CALL 768
184  XDRAW  ASC (PT$) - 31 AT X1,Y1:Y1 = Y1
      + 20
186  IF SN = 5 THEN Y1 = 70:X1 = 173:PT$ =
     ">"
188  IF SN = 10 THEN X1 = 98:Y1 = 70:PT$ =
     "<":SN = 0
190  XDRAW  ASC (PT$) - 31 AT X1,Y1
192 SN = SN + 1: RETURN
400  REM  ------------
401  REM   INITIALIZE
402  REM  ------------
405 SN = 1:PT$ = "<":CO = 3
410 X1 = 98:Y1 = 70
415 P1 =  - 16368:P2 =  - 16384
420 D$ =  CHR$ (13) +  CHR$ (4)
425  POKE P1,0
430  PRINT D$;"BLOAD SOUND"
435  PRINT D$;"BLOAD ALPHA-TBL"
440  POKE 232,168: POKE 233,97
445  HGR2 : SCALE= 1: HCOLOR= 3: ROT= 0
450  POKE 0,70: POKE 769,5
455  RETURN
500  REM  ----------
510  REM    DATA
515  REM  ----------
520  DATA  "MATCH","PIANO","PAIRS","CATCH",
     "BOXES"
530  DATA  "POKER","EQUATE","SPIRAL","LANDE
     R","SPIDER","GAMES"
535  DATA  "GAME*NOT*AVAILABLE"
540  DATA  "GRAPHIC GAMES MENU TERMINATED"
550  DATA  "APPLE GRAPHICS GAMES"
560  DATA  "RET/PLAY"
570  DATA  "SPACE/SELECT"
580  REM  --- READ ARRAYS --
```

```
585  FOR I = 1 TO 16: READ WD$(I): NEXT
599  RETURN
600  REM  ------------
601  REM     SETUP
602  REM  ------------
612  HGR2 : HGR : POKE  - 16302,0
614 WD = 14:LX = 40:LY = 50: GOSUB 900
616  POKE 28,180
618  POKE 230,64: CALL 62454: HCOLOR= 0: GOSUB
     630
620  POKE 230,32: POKE 28,180: CALL 62454
624  POKE 230,32: HCOLOR= 0: GOSUB 630
628  POKE 230,64: GOTO 650
630  FOR I = 1 TO 101
635  HPLOT 139 + I,20 TO 139 + I,191: HPLOT
     140 - I,20 TO 140 - I,191
640  NEXT
645  RETURN
650  REM  -- PAGE 2 DRAW --
652  HCOLOR= 3
654  FOR I = 1 TO 5: HPLOT 39,30 + I TO 241
     ,30 + I: NEXT
656  HCOLOR= 5
658  FOR I = 1 TO 20: HPLOT 39,35 + I TO 24
     1,35 + I: NEXT
660  HCOLOR= 3
662  FOR I = 1 TO 5: HPLOT 39,55 + I TO 241
     ,55 + I: NEXT
664  HCOLOR= 0: FOR I = 1 TO 12: HPLOT 112,
     40 + I TO 165,40 + I: NEXT : HCOLOR= 3

666 WD = 11:LX = 116:LY = 50: GOSUB 900
668 LY = 85:IC =  - 3
670  FOR J = 1 TO 10:LX = 50
672 WD = J
674  IF J > 5 THEN LX = 180
```

```
676  IF J = 6 THEN LY = 73:IC = 3
678  GOSUB 900:LY = LY + 20: NEXT
680 IC = 0
684 WD = 15:LX = 100:LY = 180: GOSUB 900
686 WD = 16:LX = 79:LY = 191: GOSUB 900
700  REM  --- FLIP PAGES ---
705  HCOLOR= 3
710  HPLOT 39,191 TO 105,165
715  HPLOT 105,165 TO 105,60
720  HPLOT 105,60 TO 170,60 TO 170,165 TO 2
     40,191
725  HPLOT 105,165 TO 170,165
730  FOR I = 1 TO 250: POKE  - 16300,0: POKE
      - 16299,0: NEXT
778  REM  -- DRAW POINTER --
779  XDRAW  ASC (PT$) - 31 AT X1,Y1: RETURN
800  REM  -- RUN PROGRAM --
805  ONERR  GOTO 815
810  GOTO 825
815  IF  PEEK (222) = 6 THEN GOSUB 850: GOTO
     100
820  TEXT : PRINT "ERROR....."; PEEK (222):
      END
825  POKE 800,1
830  PRINT D$;"RUN ";WD$(SN)
850  REM  -- ERROR MESSAGE --
855 CO = 3:WD = 12:LX = 50:IC = 0:LY = 30: GOSUB
     900: POKE 216,0
860  FOR T = 1 TO 50: NEXT : POKE P1,0
865 CO = 0:WD = 12:LX = 50:LY = 30:IC = 0: GOSUB
     900:CO = 5
870  RETURN
899  END
900  REM  ---------------
905  REM    JPRINT WORDS
910  REM  ---------------
```

```
915  HCOLOR= CO:LH = LY
920  FOR I = 1 TO  LEN (WD$(WD))
925 SV$ =  MID$ (WD$(WD),I,1):SV =  ASC (SV
     $) - 31
930  DRAW SV AT LX,LY
935  IF WD = 14 THEN  FOR T = 1 TO 30: NEXT
     : CALL 768
940 LX = LX + 10
945 LY = LY + IC
950  NEXT :LY = LH
955  RETURN
```

變數使用說明

Z……按鍵的ASCII碼。

P1,P2 …鍵盤及鍵盤清除位址。

X1,Y1 …選擇指標的座標位置。

DT$ ……遊戲名稱字串。

IG ……LY 的垂直增量。

SN ……你所選擇的遊戲的號碼。

D$ ……RETURN + Control – D字元。

LH ……儲存 LY 變數。

LX,LY …字串顯示副程式的座標位置。

I,J ……迴路變數。

WO ……在字串矩陣變數中的字串號碼。

矩陣使用說明

WD$(14)……文字字串。

遊戲鍵說明

空格鍵……移動選擇指標。

RETURN ……輸入你所要的遊戲。

ESC ……結束節目單程式。

程式說明

10—80行 ……宣布矩陣。

90—95行 ……副程式。

100—135行 ……輸入迴路，並檢查鍵盤是否被按下。

180—192行 ……移動指標副程式：消去舊的指標，繪出新的指標，並且設定 SN 的增加值。

400—455行 ……設定初值，輸入聲音及圖形副程式。

500—599行 ……把資料讀入矩陣之中。

600—645行 ……設定 menu 畫面，清除兩頁高解析圖形(HGR,HGR2)並設定爲全螢幕(POKE—16302,0)。

650—678行 ……在第二頁顯示圖形。

700—730行 ……設定畫面，並且閃動第一頁及第二頁交顯

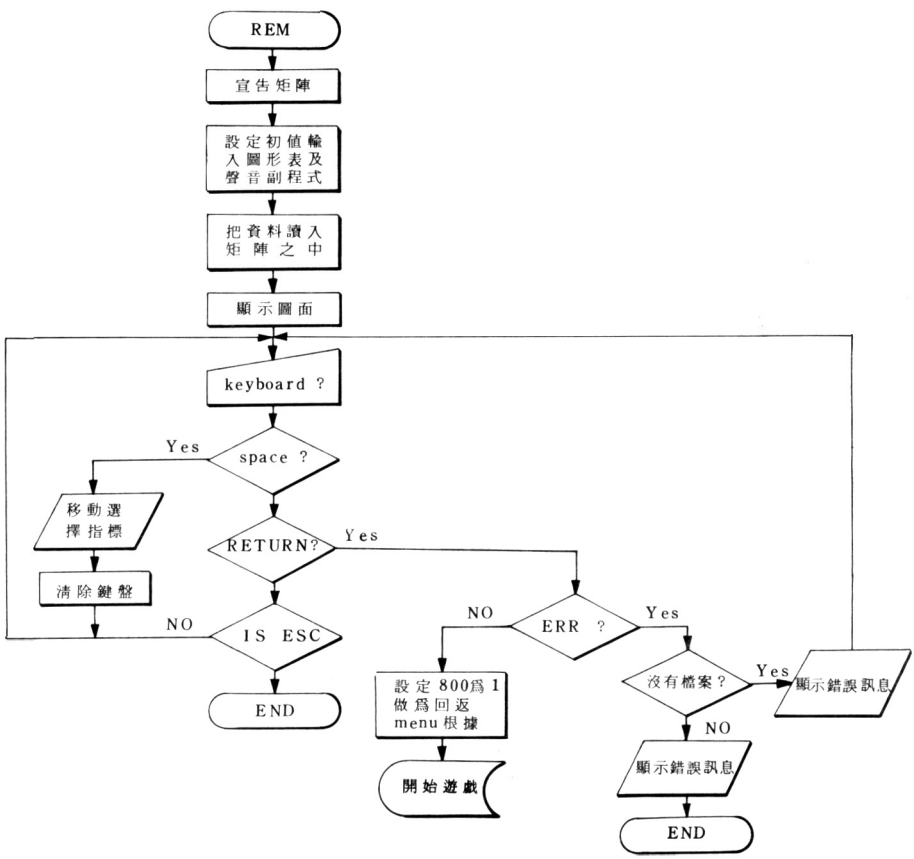
示。

800—810行 ……設定錯誤須號。假如錯誤的話，顯示錯誤訊息，否則就開始你所選擇的遊戲。

850—899行 ……錯誤訊息副程式。

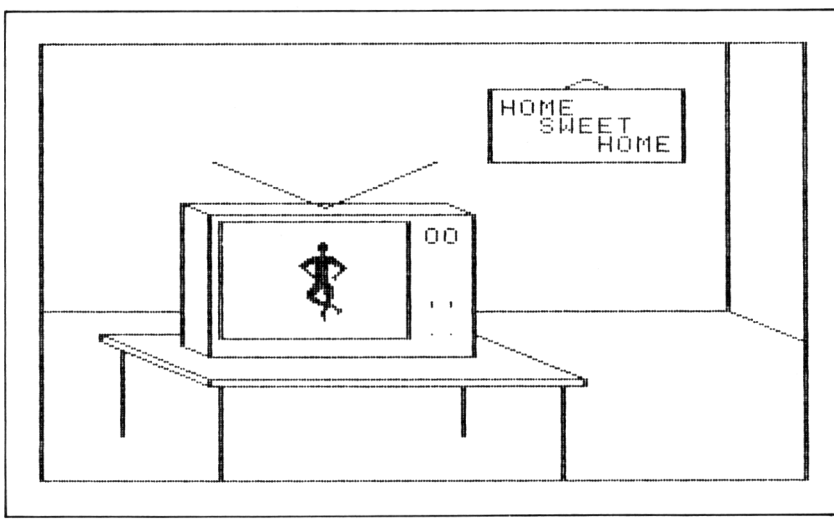
900—950行 ……字形顯示副程式。

主程式流程圖



第18章

創造自己的GAME



幾乎這本書所有的程式，均採用了同樣的字形圖形表及聲音副程式，你可以將它當成兩套基本工具來使用，所以你只要造出程式所要用到的特殊圖案的圖形表，然後專注於遊戲情節的安排便可以了，這樣子，程式的設計工作便得以大大的簡化。

對於聲音副程式的使用，你可以將 0～255 的數 POKE 到 0 這個位址，然後 CALL 768 便可以造出各種音高的聲音，而將 0～255 的數 POKE 到 1 這個位址，CALL 768 之後，便可以造出各種長度的聲音來。將第 10 章 PIANO 的演奏副程式

附加到你的Game 裡頭，更可以造出美妙的音樂效果呢！

至於字形顯示，我們也在第三章提過，使用程式3－的架構，可以任意的在高解像畫面印出美麗的文字來，第3章的英文字形表，有許多是APPLE 文字幕沒有的特殊字形或符號呢！使用時，我們提供您如下的程式架構做爲參考：

```
]LIST

0  REM ********************************
1  REM **  GAME USING ALPHA TABLE  **
2  REM ********************************
5  HGR : HCOLOR= 3: ROT= 0: SCALE= 1
6 LX = 10:LY = 10
10 N$ = "BEGIN"
20  POKE 232,168: POKE 233,97
30  GOSUB 1000
40  REM ----------------------
43  REM I    ..............   I
45  REM I    ..............   I
50  REM I    MAIN  PROGRAM    I
60  REM I    ..............   I
70  REM I    ..............   I
80  REM ----------------------
100 N$ = "  END !!!   "
110  POKE 232,168: POKE 233,97
115 LX = 180:LY = 150
120  GOSUB 1000
130  END
1000  FOR I = 1 TO  LEN (N$)
1010 SV =  ASC ( MID$ (N$,I,1))
1015 L = SV - 31: IF L < 1 THEN L = 1
1020  DRAW L AT LX,LY
```

```
1030 LX = LX + 9
1040  NEXT I
1050  RETURN
```

程式中 40 行～ 80 行代表主程式的區域（眞正的主程式當然不是這麼短了），第 5 行是用來做畫面的初始設定，請注意，ROT 及 SCALE 一定要定好一個數值，否則畫出來的字形會因放大或旋轉而亂成一團， 20 行及 110 行的 POKE 是令 APPLE 畫出位於 $61A8 開始的字形，如果在畫完字形之後，要畫出圖案，則必須，將（ 232,233 ）指位器指向圖案圖形表的起始位址才行。

接下來，便是該你來設計遊戲程式了，依我們的經驗，你可以找一個日常常見的遊戲來做練習，將其內容加以分析，畫成流程方塊，然後仿照曾解析過的程式架構來設計，慢慢的你便能夠全盤掌握運用自如了，說不定你會青出於藍而發展出更巧妙的方法呢！

參考書目

Ref. 1 APPLE Ⅱ Reference BASIC Programming Reference Manual.

Ref. 2 Collision Counter, Richard Cornelius and Melvin Zandler , Nibble／VOL.4／NO.1／1983

Ref. 3 APPLE and PET, I／O 別册 工學社（日本）出版。

Ref. 4 APPLE GRAPHICS GAMES, Paul Coletta, Reston Publishing Inc。

Ref. 5 APPLEⅡ 微電腦徹底研究㈠第四章，施威銘著，波前電腦圖書出版。

Ref. 6 APPLEⅡ 微電腦徹底研究㈡附錄，施威銘著，波前電腦圖書出版。

Ref. 7 APPLEⅡ 微電腦徹底研究㈤第一章，第二章，施威銘著，波前電腦圖書出版。

Ref. 8 マイヲソ・グーム徹底研究2，I／O別册⑨工學社（日本）出版。

Ref. 9 從APPLEⅡ 到圖形遊戲製作（GRAPHIC GAME），宗世麟著，波前電腦圖書出版。

著作權、版權所有・翻印必究

法律顧問：邱晃泉律師

APPLE Ⅱ 圖形遊戲系統設計

編 著 者：林 和 賢
出 版 者：林 和 賢
地　　址：台北縣板橋市林森街一巷十一號
總 經 銷：波前電腦管理圖書有限公司
地　　址：台北市泰順街二巷三十號
電　　話：3927101～2
郵　　撥：503611　　帳　戶：林和賢
定　　價：180元
印 刷 者：名揚印刷有限公司
中 華 民 國　73 年　4 月 出 版

波前電腦提供您磁片服務

■ APPLE Ⅱ世界程式競賽選輯
（特價 170 元）

- 全部十餘程式，省掉KEY - IN時間
- 世界程式競賽專輯
- 星艦迷航記系列程式

■ APPLE Ⅱ徹底研究㈤磁片　動畫技巧篇
（特價 250元）

- 專業的技法，動畫的極至，附使用手冊。
- 包括全書所有程式(改良版)，附LISA組譯器。
- 使用本磁片，便可以寫出自己的 GAME 來。

■ APPLE Ⅱ徹底研究㈢磁片　磁碟系統篇
（特價 170元）

- 包含 ZAP、FIXCAT 及全書所有程式

■ APPLE Ⅱ圖形遊戲系統設計　磁片
（特價 250元）

- 包含大蜘蛛、金樸克、……等十個GAME。

此欄係備寄款人與帳戶通訊之用，惟所作附言應以關於該次劃撥事項爲限。

通信欄
茲向貴公司購買 □ APPLE II 徹底研究㈣磁片（250 元） □ APPLE II 徹底研究㈡磁片（170 元） □ APPLE II 圖形遊戲系統設計磁片（250 元） □ APPLE II 世界程式競賽選輯磁片（170 元） □ APPLE II 徹底研究㈠及㈡共一片（170 元） 右列價格均包含掛號郵費 寄款人：

局號：　　收據號碼：

郵政劃撥儲金存款單

收款帳號	寄款人姓名住址	新臺幣
0-503611-7		（請用大寫數目字填寫並於數末加一整字）
收款戶名		
板橋市林森街一巷十一號 林和賢		中心局郵戳
經辦局郵戳		
	手續費　次　元	

請注意：一、帳號、戶名及寄款人姓名住址請詳細填明，以免誤寄。
二、抵付交換票據之存款，務請於交換前一、二天存入，必要時，可請存款局先以電話通知劃撥儲金組，惟長途電話費由存款人負擔。

主管：　　經辦：

本聯由劃撥儲金組存查

郵政劃撥儲金存款通知單

收款帳號	寄款人姓名住址	新臺幣
0-503611-7	姓名： 地址： 電話：	（請用大寫數目字填寫並於數末加一整字）
收款戶名		
板橋市林森街一巷十一號 林和賢		經辦局郵戳
	手續費　次　元	

主管：　　經辦：

本聯經劃撥儲金組登帳後寄交寄戶帳

■存款後由郵局掣給正式收據爲憑，本單不作收據用■

封面設計／施威銘
美工完稿／嚴　珣

介紹您：波前的新書

BASIC學習者的一大步

Apple II組合語言技術(即將出版)　施威銘著
Apple II 組合語言精華　李天佑著

DOS徹底研究

Apple II徹底研究(三)　施威銘著

本書的姐妹作

世界程式競賽選輯　林和賢著

GAME極限的挑戰

Apple II 徹底研究(五)　施威銘著
(動畫技巧篇)

・波前出版・必屬好書・

本書介紹您……

- 系統化的遊戲程式設計。
- 眞正實用的自動造表程式。
- 簡單有效的音樂程式。
- 只用BASIC便可以造出高級的GAME。
- 大蜘蛛等十則遊戲，篇篇精彩。
- 所有程式，提供磁片服務。

波前電腦管理圖書有限公司

MCA-011

APPLE II 圖形遊戲系統設計

施威銘 校訂

林和賢 編著

波前